IEEE ANTS™
15–18 December 2024
Assam, India
IIT Guwahati

UNIVERSITÀ DEGLI STUDI DELL'AQUILA
॥ सा विद्या या विमुक्तये ॥
ಭಾ.ತಂ.ಸಂ. ಧಾರವಾಡ
भा. प्रौ. सं. धारवाड
IIT DHARWAD
GHENT UNIVERSITY
THE UNIVERSITY OF TEXAS AT DALLAS
UTD
EST. 1969

EPIC-RAN: **E**valuating **P**ower **i**n **C**ontainer Clouds for Open **R**adio **A**ccess **N**etworks

Venkateswarlu Gudepu, Rajashekhar Reddy Tella, **Carlo Centofanti**, José Santos, Andrea Marotta

Speaker

Carlo Centofanti

Assistant Professor
University of L'Aquila, Italy
Carlo.centofanti1@univaq.it

Outline

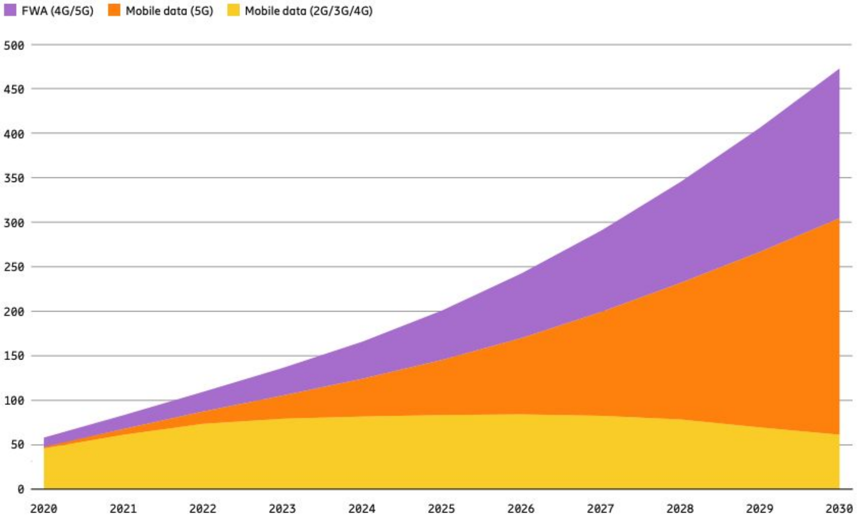
- **Introduction to Energy Consumption in RANs**
  - Challenges and impact on sustainability
  - State-of-the-art energy measurement tools
- **RAN Evolution and Virtualization**
  - From Legacy RAN to Open RAN
  - Monolithic, Disaggregated, and CUPS architectures
- **Containerized RANs and Energy Efficiency**
  - Containers vs Virtual Machines
  - Kubernetes for orchestration and monitoring
- **Energy Monitoring Tools**
  - Hardware-based: Smart Plugs, Professional Power Meters
  - Software-based: Scaphandre, Kepler, S-tui
- **Hands-on Demonstration**
  - Energy measurement setup for Monolithic, Disaggregated, and CUPS RANs
  - Grafana integration and visualization
- **Key Learnings and Future Directions**
  - Insights, open challenges, and sustainability strategies

Motivations

Introduction

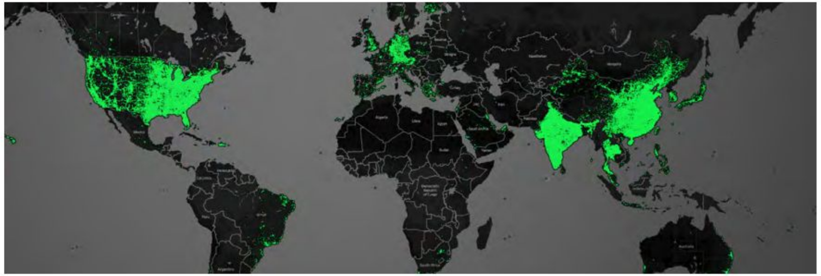
80 percent of mobile data traffic globally in 2030

Figure 5: Global mobile network data traffic (EB per month)



5G SA Deployment coverage

Figure 13: 5G SA deployment and trials – indicative coverage



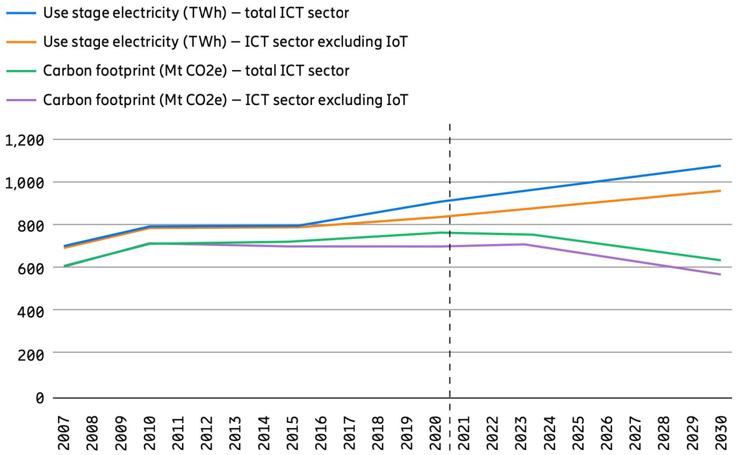
Source: Based on analysis by Ericsson of Ookla® Speedtest Intelligence® data for January 2022 to October 2024. Note: Samples include iOS and Android smartphones connecting to a 5G SA network. Sample density varies across markets, reflecting differences between markets with more extensive commercially launched 5G SA or markets with mainly trials/tests.

Introduction

ICT sector 2007-2030 electricity use and carbon footprint

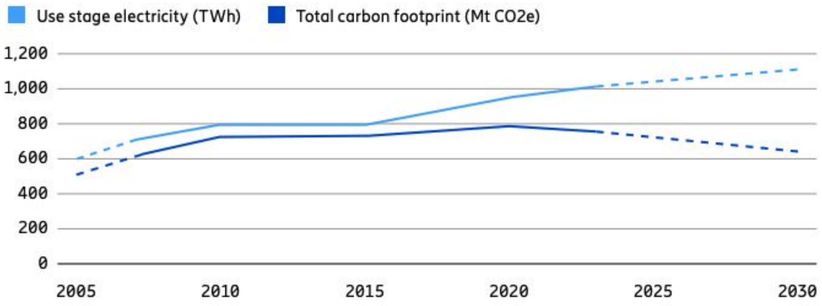
Predictions from 2023 Vs now in 2024

Figure 25: ICT sector development 2007–2020 and forecast to 2030



2023

Figure 31: ICT sector development 2007–2023 and forecast to 2030

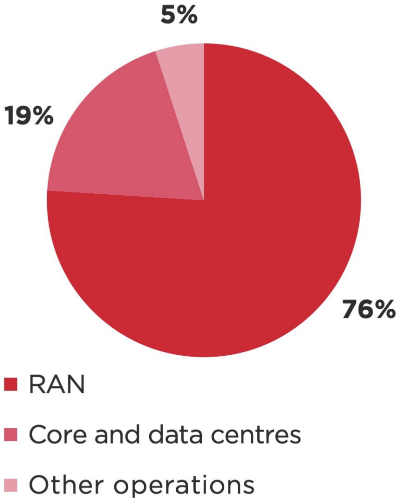


2024

Introduction

Figure 4

Where mobile operators use energy in their network operations

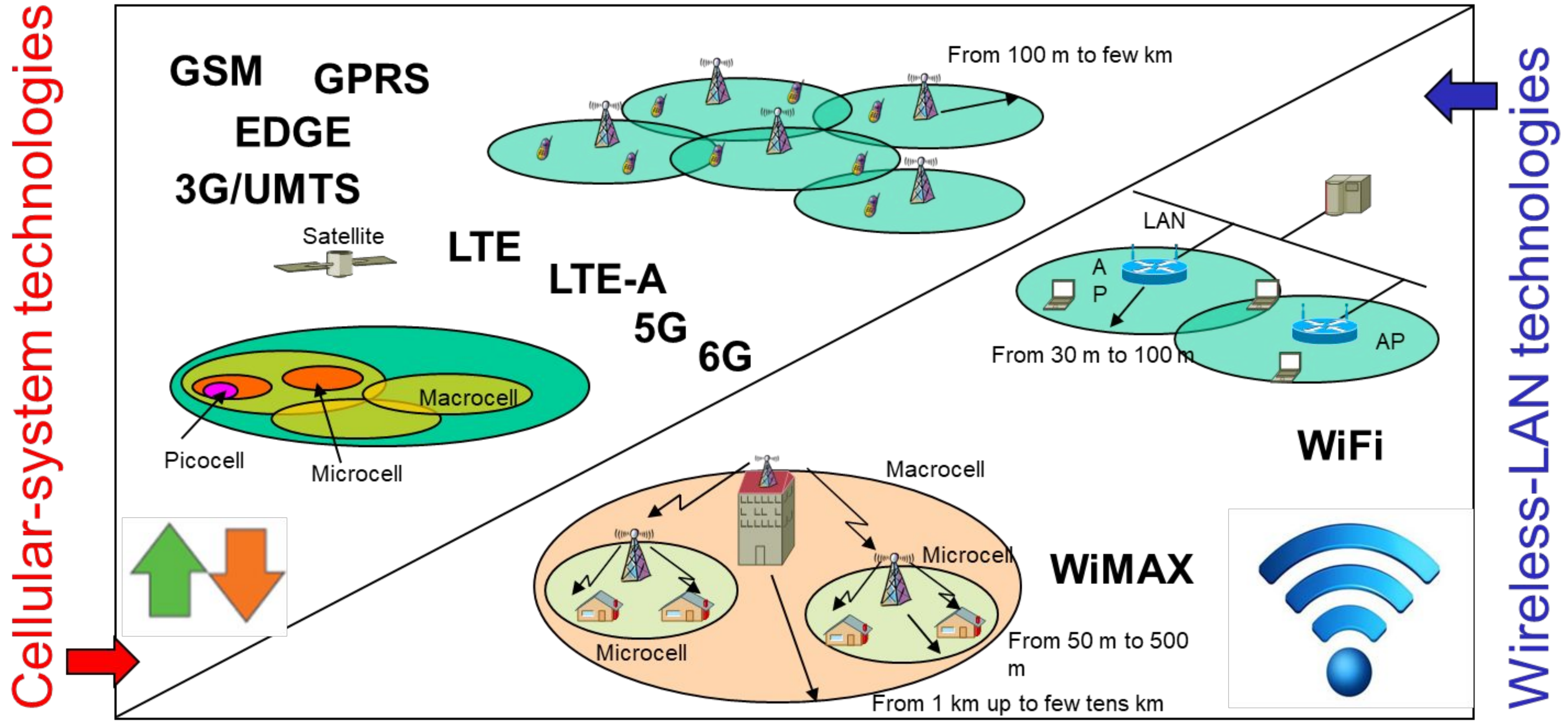


Source: GSMA Intelligence

Note: Data based on GSMA Intelligence's Mobile Energy Efficiency Benchmarking project in 2024 with 65 mobile networks.

Overview of Radio Access Networks (RANs) and Evolution

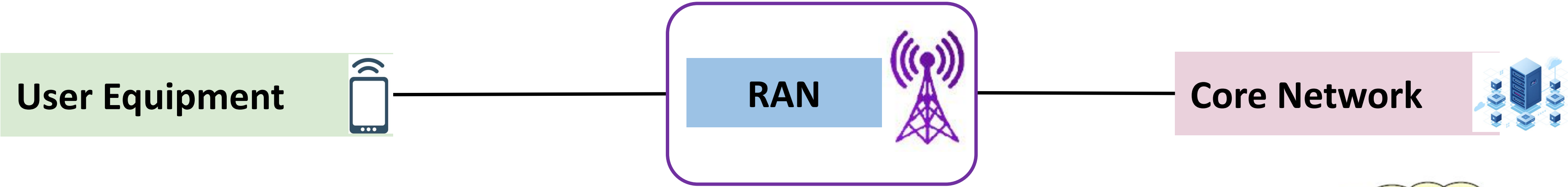
Overview of Radio Access Networks (RANs) and Evolution (1)



Overview of Radio Access Networks (RANs) and Evolution (2)

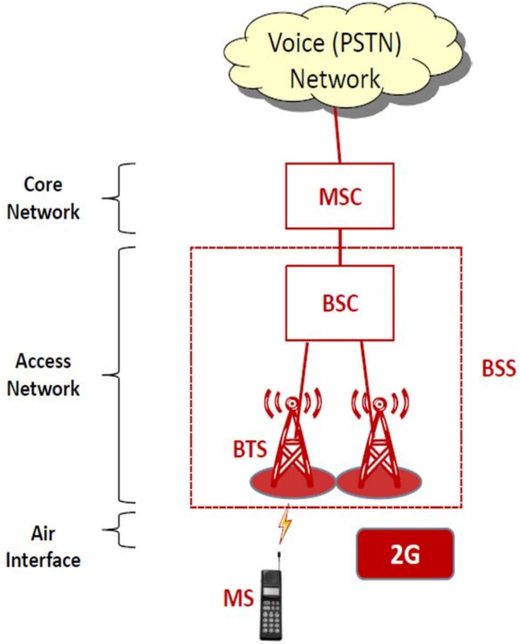
Cellular-system technologies	Wireless-LAN technologies
Designed for extensive geographic coverage, including urban, suburban, and rural areas.	Primarily designed for small geographic areas such as homes, offices, and campuses.
Supports seamless handover and roaming, allowing users to move across large areas without losing connection.	Provides connectivity within the range of a single access point, with limited support for seamless handover.
Operates on licensed frequency bands regulated by government agencies, ensuring minimal interference.	Operates on unlicensed frequency bands (e.g., 2.4 GHz and 5 GHz), which can lead to potential interference.
Requires a network of base stations (cell towers) and centralized control for communication.	Utilizes access points and routers, typically managed locally by users or IT staff.
Includes technologies like 3G, 4G LTE, and 5G.	Includes technologies like Wi-Fi (802.11a/b/g/n/ac/ax).

Overview of Radio Access Networks (RANs) and Evolution (3)



2nd Generation (2G):

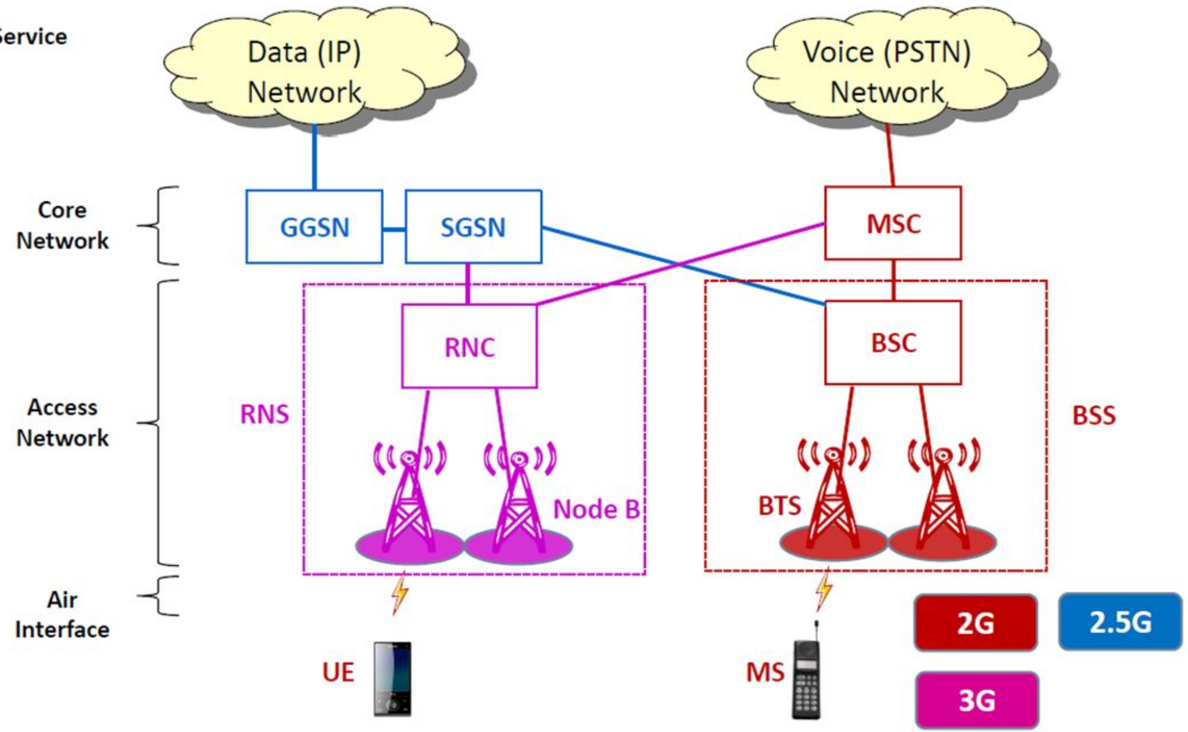
- The Mobile Station (MS) consists of two parts:
  - The phone (IMEI=International Mobile Equipment Identity)
  - SIM card (IMSI=International Mobile Subscriber Identity)
- Base Station Subsystem (BSS) (a BSC + several BTS)
  - The GSM RAN is a set of BSS
- MSC ⇒ handles communication routing with other networks in connection with DBs: VLR, HLR.
- BTS: Base Transceiver Station (or base station)
  - a BTS per cell (in center-excited cells)
  - each BTS has many TRX, one per frequency used in that cell
- BTS handles physical layer (modulations, frequency management and slot timing)



Universal Mobile Telecommunications System (UMTS) (3G):

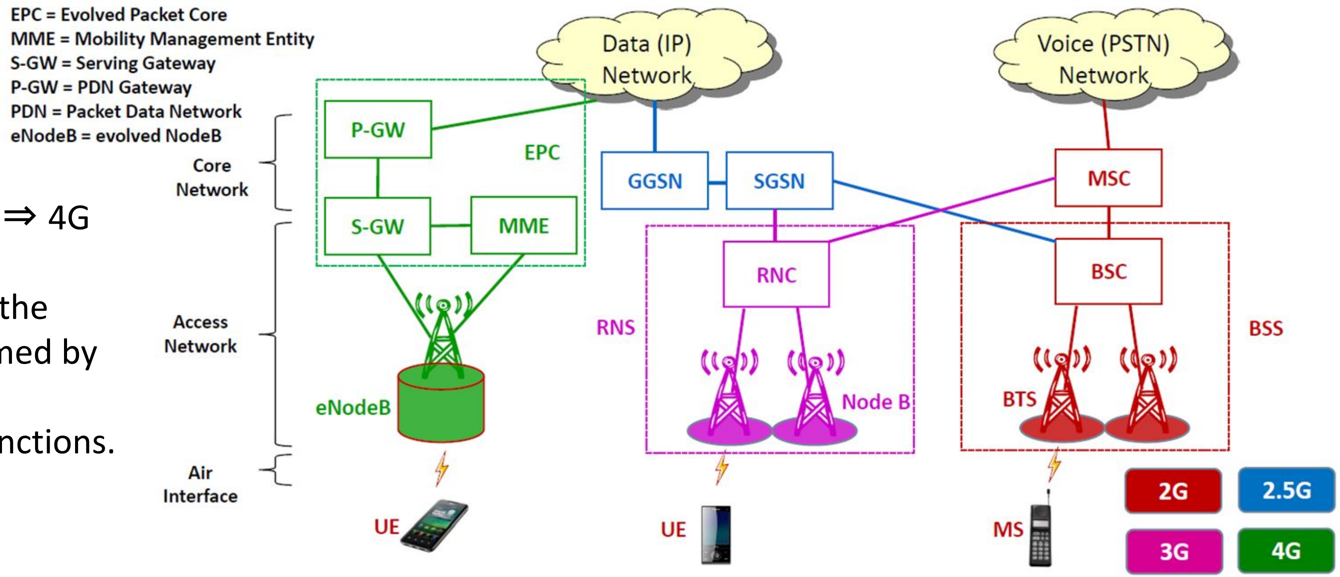
UMTS = Universal Mobile Telecommunications Service
RNC = Radio Network Controller
RNS = Radio Network Subsystem
UE = User Equipment

- RNC carries out the resource management and mobility management functions controlling Node B
- Node B has similar physical layer functions as the BTS but some functions are off-loaded



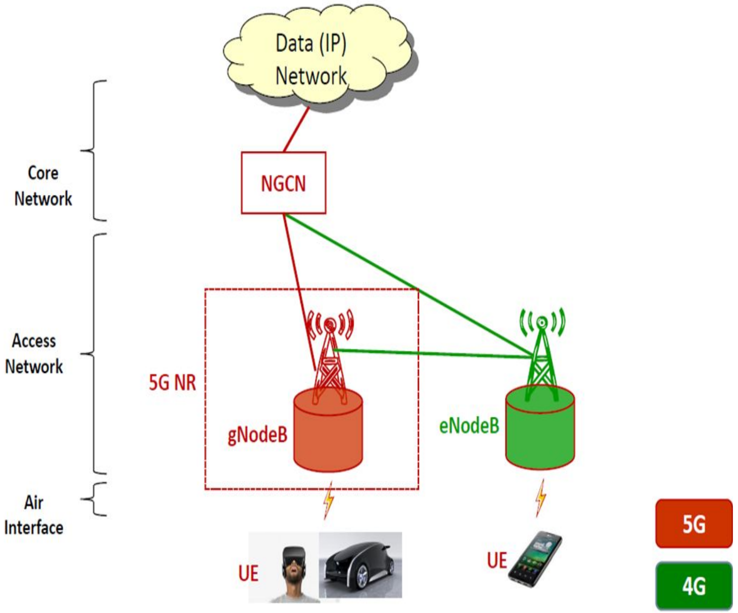
LTE Architecture (4G)

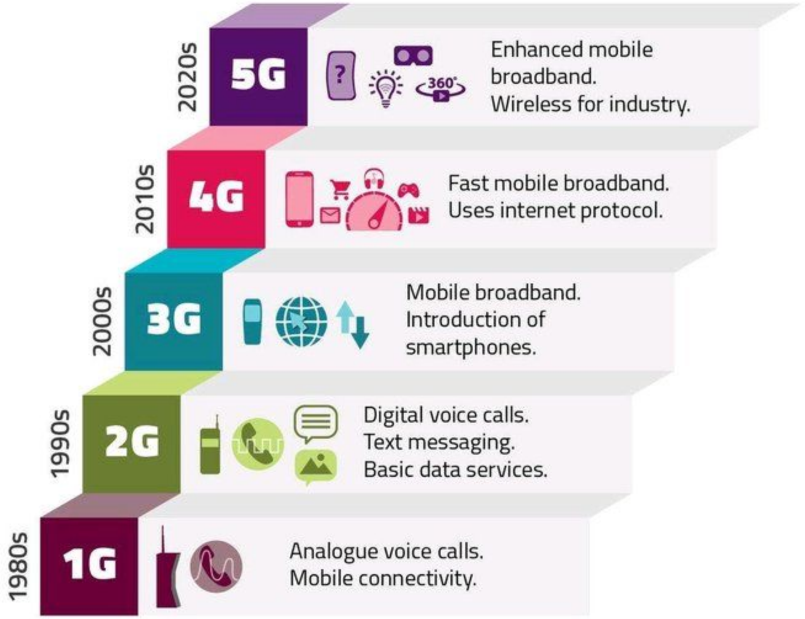
- eNodeB (evolved NodeB) ⇒ 4G BTS.
- eNodeB ⇒ Absorbs back the functions that are performed by the 3G RNC in addition to standard physical layer functions.



5G RAN

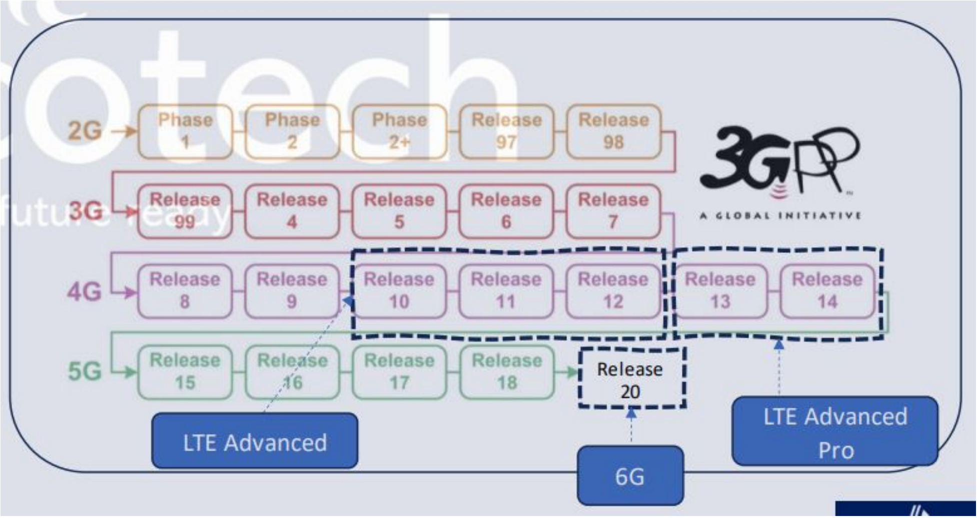
NGCN = Next Generation Core Network
gNodeB = next generation NodeB
NR = New Radio





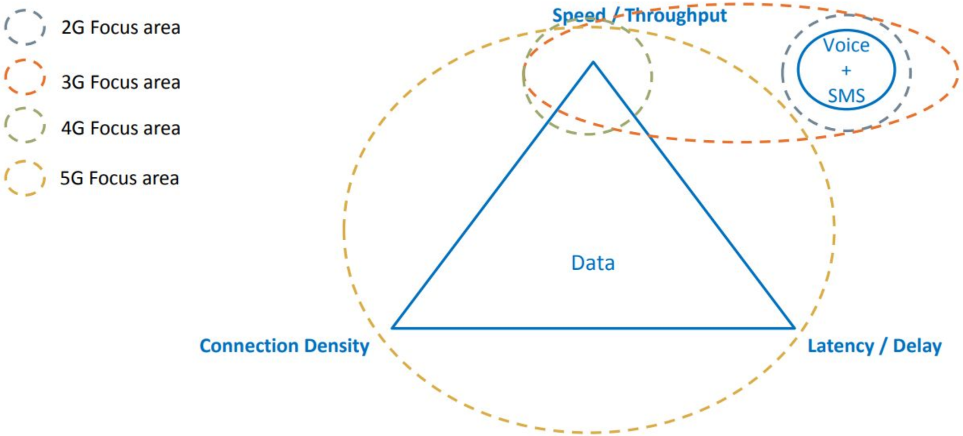
5G RAN

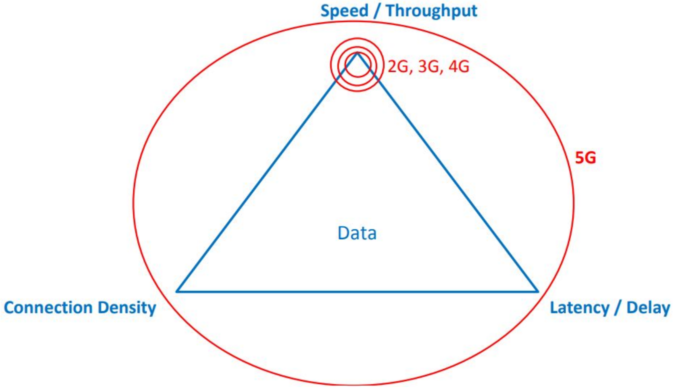
- 3GPP keeps on releasing new specifications quite often with new features and use cases.
- 4G related specifications starts from Release 8
- 5G including Private Networks (Non-Public Network) related specifications starts from Release 15
- 6G is expected to start from Release 20

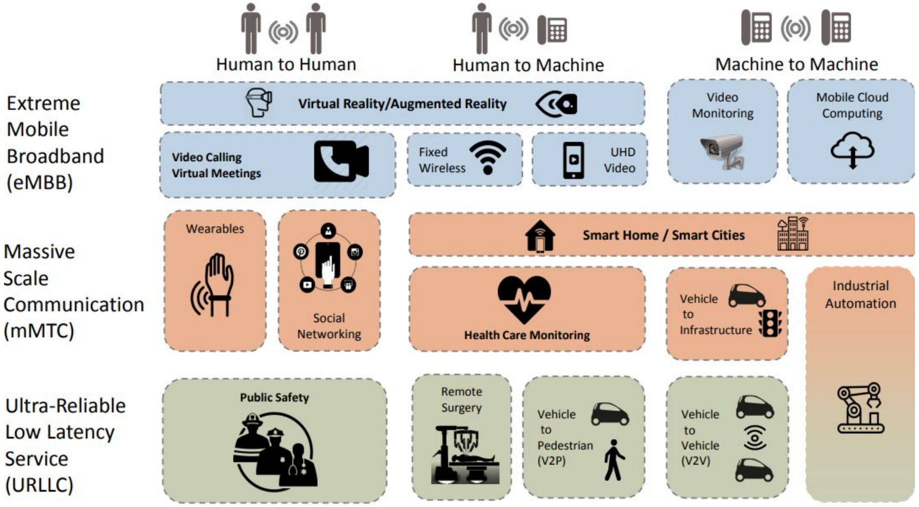


Source: Infotech

5G Focus area



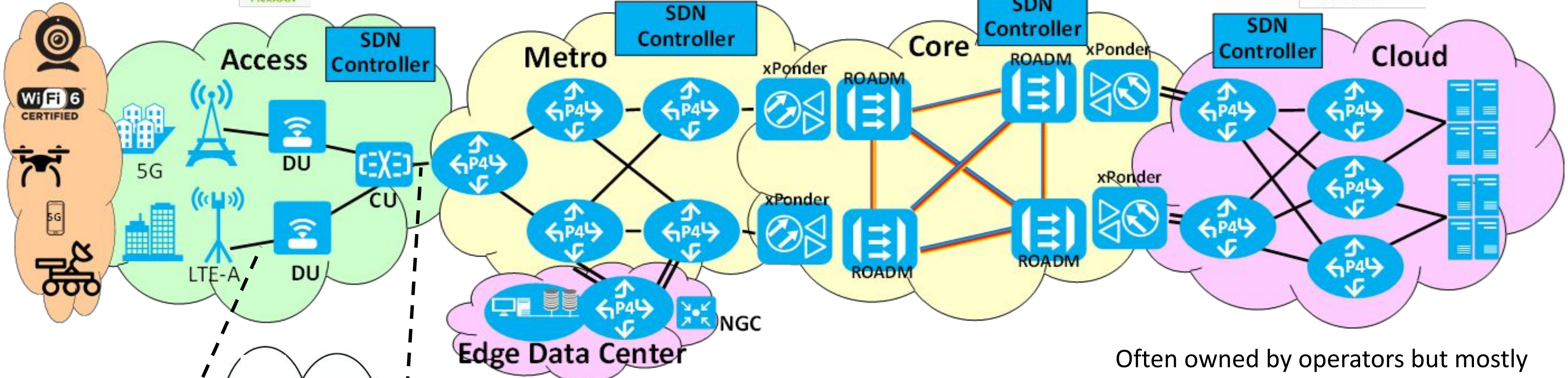


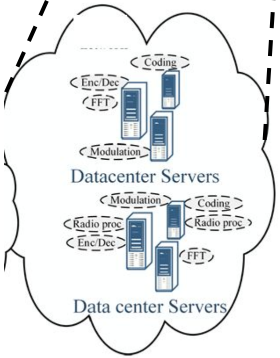


5G RAN deployment scenarios to meet various use cases:

1. Monolithic
2. Disaggregated
3. Control User Plane Separation (CUPS)

A wide picture of the Network



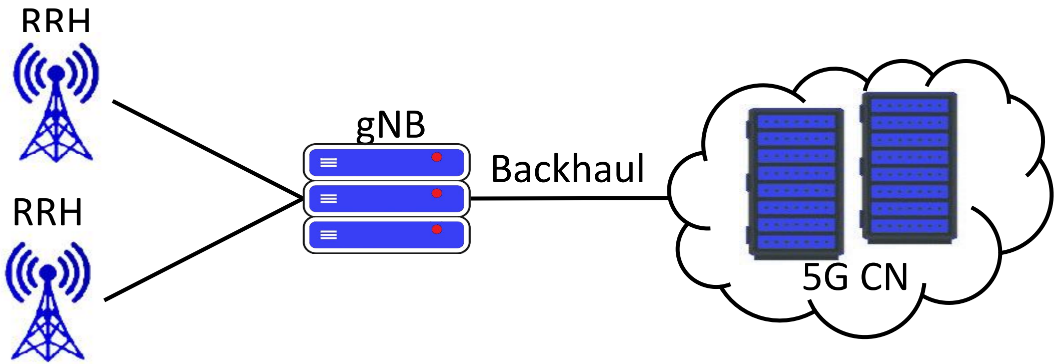


For services requiring real-time or low-latency In-network computation

Often owned by operators but mostly by over-the-top corporations

Data Centers exclusively devoted for mo-demodulation and separation of traffic coming from the mobile interface

5G RAN: Monolithic



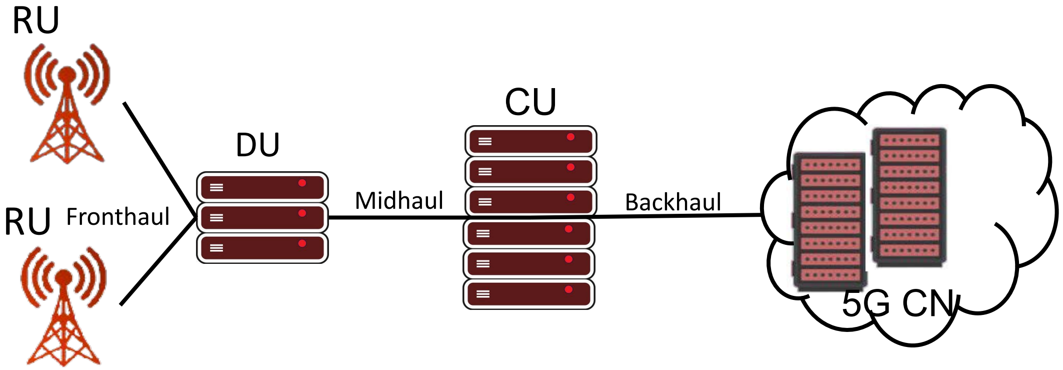
RRH - Remote Radio Head; BBU - Base Band Unit; 5G CN - 5G Core Network

- Simplicity of Design
- Low Latency
- Reliability
- Cost-effective for small deployments
- Ease of maintenance
- Energy efficiency

- Combines all the RAN functionalities into a Next-generation NodeB (gNB)
  - Physical
  - Medium Access Control (MAC)
  - Radio Link Control (RLC)
  - Packet Data Convergence Protocol (PDCP)
  - Radio Resource Control (RRC)

- Limited scalability
- Vendor lock-in
- High cost for large deployments
- Inflexibility
- Inefficient resource utilization

5G RAN: Disaggregated



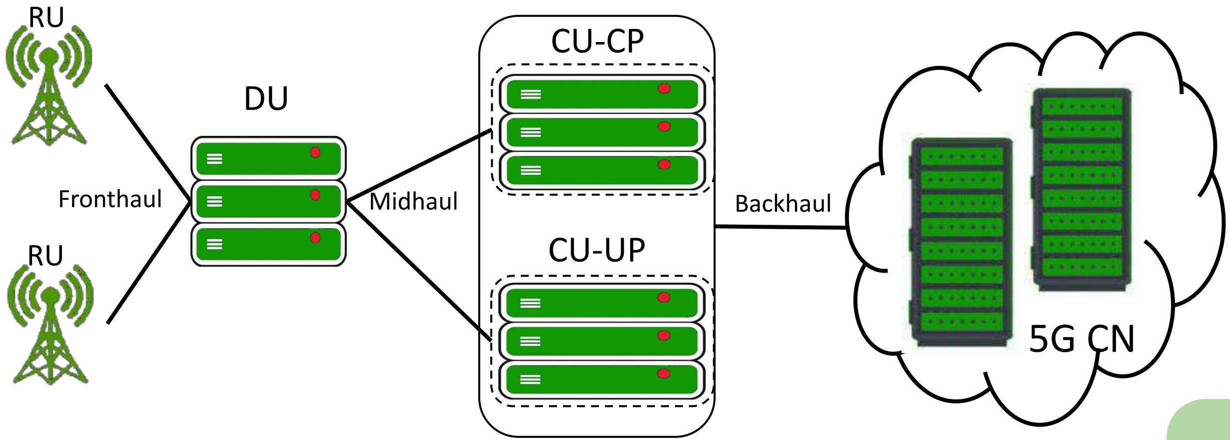
RU - Radio Unit; DU - Distributed Unit; CU - Centralised Unit; 5G CN - 5G Core Network

- Flexibility
- Scalability
- Resource Optimization
- Cloud-Native Deployments

- The gNB functions are physically decoupled
  - Radio Unit (RU)
    ⇒ PHY and Radio Frequency (RF) functionalities
  - Distributed Unit (DU)
    ⇒ MAC and portions of the PHY layer
  - Central Unit (CU)
    ⇒ RRC and PDCP

- Increased Complexity
- Latency Concerns
- Management Overhead

5G RAN: Disaggregated



RU - Radio Unit; DU - Distributed Unit; CU-CP - Centralised Unit Control Plane; CU-UP - Centralised Unit User Plane; 5G CN - 5G Core Network

- gNB-CU further splits into
  - gNB-CU-CP ⇒ Responsible for the initial attachment procedure between the RAN components
  - gNB-CU-UP ⇒ To forward the data packets

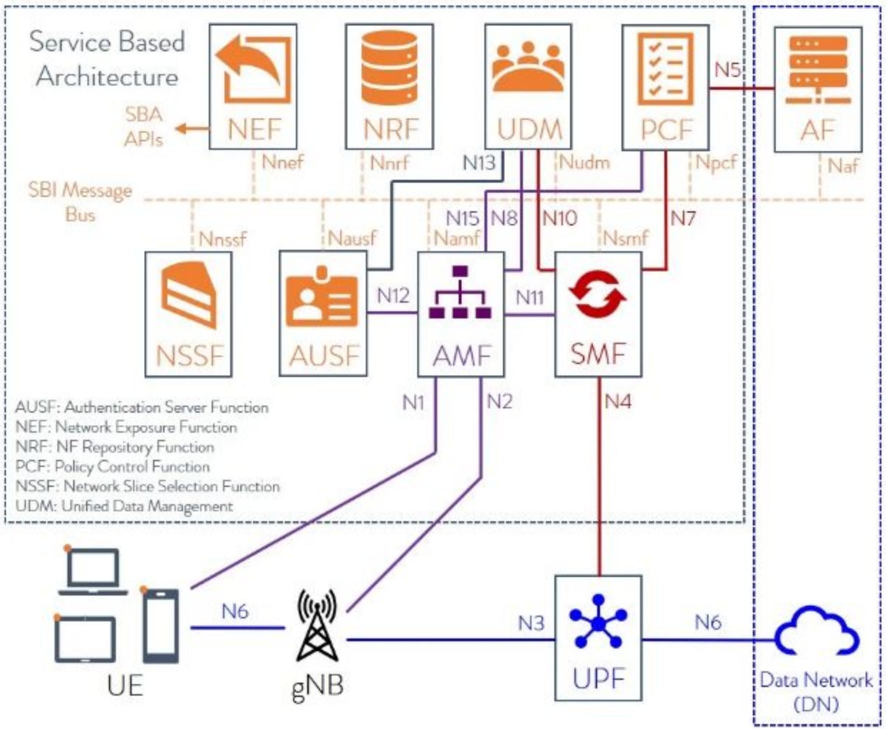
- Better scalability of UP operations
- Reducing OPEX and CAPEX for MNOs

- Management Overhead

The 5G SBA Architecture

5G Core SBA

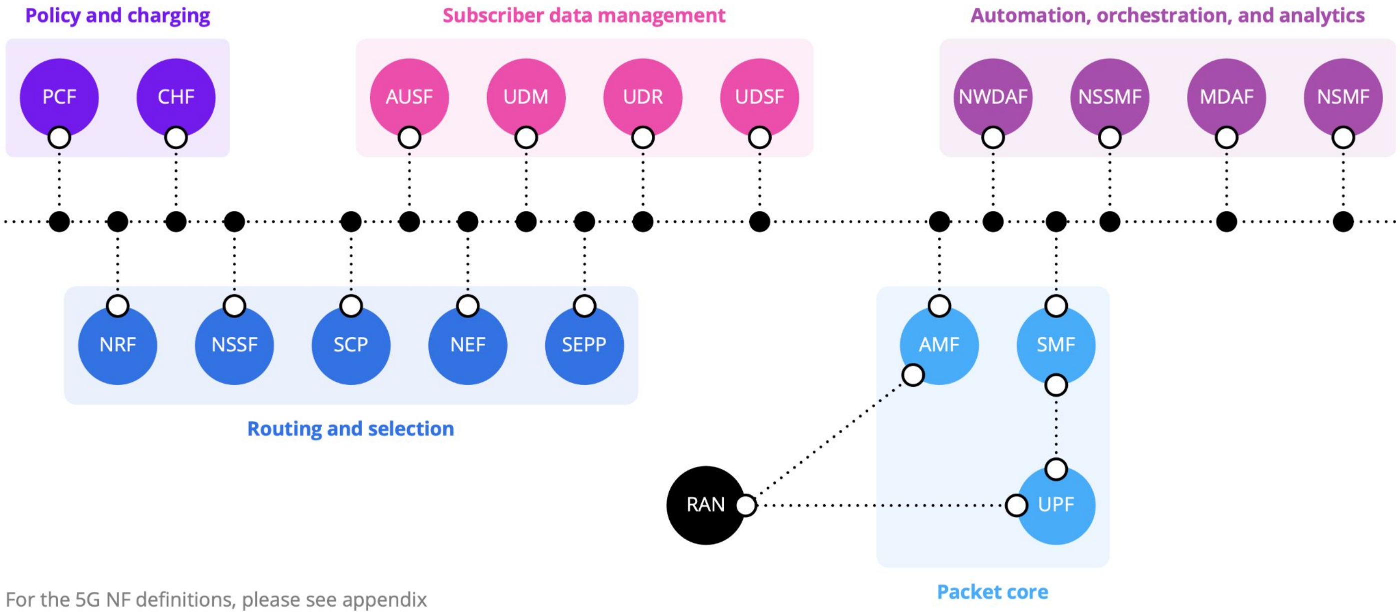
NETWORK FUNCTION	DEFINITION	DESCRIPTION
NEF	Network exposure function	This NF exposes the 5G network's capabilities and services through established application programming interfaces (APIs), facilitating interaction with third parties, such as application developers.
NRF	Network repository function	The NRF provides a record of all NFs available on the platform, together with a profile of each and the services they support. When a new NF is brought up, it registers its IP address and capabilities with the NRF. A consumer NF will request the NRF to identify which NFs are registered with it and have the required capabilities and can act as a producer of services. It interacts with the SCP.
UDM	Unified data management	The UDM is a central repository of subscriber data including the subscriber profile, authentication data, and other service-related information. It ensures that user data sessions are set up and torn down correctly. A stateful UDM stores this data locally, while a stateless version stores it in the UDR. It interacts with other NFs, such as the AUSF and NRF.
PCF	Policy control function	The PCF considers device profile, subscription information, and real-time information to enforce rules for traffic steering, QoS, and charging. It also relies on information from other NFs, such as the network exposure function (NEF) and SMF.
NSSF	Network slice selection function	The NSSF analyzes the requirements of a UE and matches those requirements with a network slice. In today's network slicing, the slices are predefined rather than autonomously orchestrated and for this, this NF must find the best match. The NSSF interacts with AMF, SMF, and PCF.
AUSF	Authentication server function	The AUSF performs the authentication of user equipment (UE) as it connects, at switch-on, or during a handover procedure from a 4G network to a 5G network. It does so by ensuring the subscriber information is transmitted and stored securely. The AUSF stores authentication keys and provides the AMF with the necessary authentication services.
AMF	Access and mobility management function	Supports the termination of control plane signaling and carries out registration and authentication of devices based on predefined policies, preventing unauthorized access. It also carries out mobility management, facilitating handover procedures as the device moves between antenna masts, to ensure an uninterrupted service.
SMF	Session management function	Handles session management and interacts with the decoupled data plane by creating, updating, and removing protocol data unit (PDU) sessions and managing IP session context within the UPF. Policy and charging control rules from the PCF are also fed as templates into the UPF, for it to deliver quality of service (QoS).
UPF	User plane function	It processes user packet data, facilitating forwarding, routing and packet inspection as well as QoS handling. It interacts with the SMF and PCF and it is an evolution of the 4G control and user plane separation (CUPS).



https://techcommunity.microsoft.com/blog/azureforoperatorsblog/what-is-the-5g-service-based-architecture-sba/3831367

5G Core Service-Based Architecture (SBA)

FIGURE 15
The 5G core service-based architecture



For the 5G NF definitions, please see appendix

SOURCE: OMDIA, ADAPTED FROM 3GPP R17 TS23.501

5G Core Service-Based Architecture (SBA)

NETWORK FUNCTION	DEFINITION	DESCRIPTION
AMF	Access and mobility management function	Supports the termination of control plane signaling and carries out registration and authentication of devices based on predefined policies, preventing unauthorized access. It also carries out mobility management, facilitating handover procedures as the device moves between antenna masts, to ensure an uninterrupted service.
SMF	Session management function	Handles session management and interacts with the decoupled data plane by creating, updating, and removing protocol data unit (PDU) sessions and managing IP session context within the UPF. Policy and charging control rules from the PCF are also fed as templates into the UPF, for it to deliver quality of service (QoS).
UPF	User plane function	It processes user packet data, facilitating forwarding, routing and packet inspection as well as QoS handling. It interacts with the SMF and PCF and it is an evolution of the 4G control and user plane separation (CUPS).
AUSF	Authentication server function	The AUSF performs the authentication of user equipment (UE) as it connects, at switch-on, or during a handover procedure from a 4G network to a 5G network. It does so by ensuring the subscriber information is transmitted and stored securely. The AUSF stores authentication keys and provides the AMF with the necessary authentication services.
UDM	Unified data management	The UDM is a central repository of subscriber data including the subscriber profile, authentication data, and other service-related information. It ensures that user data sessions are set up and torn down correctly. A stateful UDM stores this data locally, while a stateless version stores it in the UDR. It interacts with other NFs, such as the AUSF and NRF.
UDR	Unified data repository	This is the database and repository of user-related data, including user profiles and application data. It stores the stateless information of the UE. It interacts with the policy control function (PCF) to ensure the correct subscriber quality of service (QoS) and charging policy information is obtained and enforced.
UDSF	Unstructured data storage function	The UDSF supports the storage and retrieval of unstructured data from other NFs.

5G Core Service-Based Architecture (SBA)

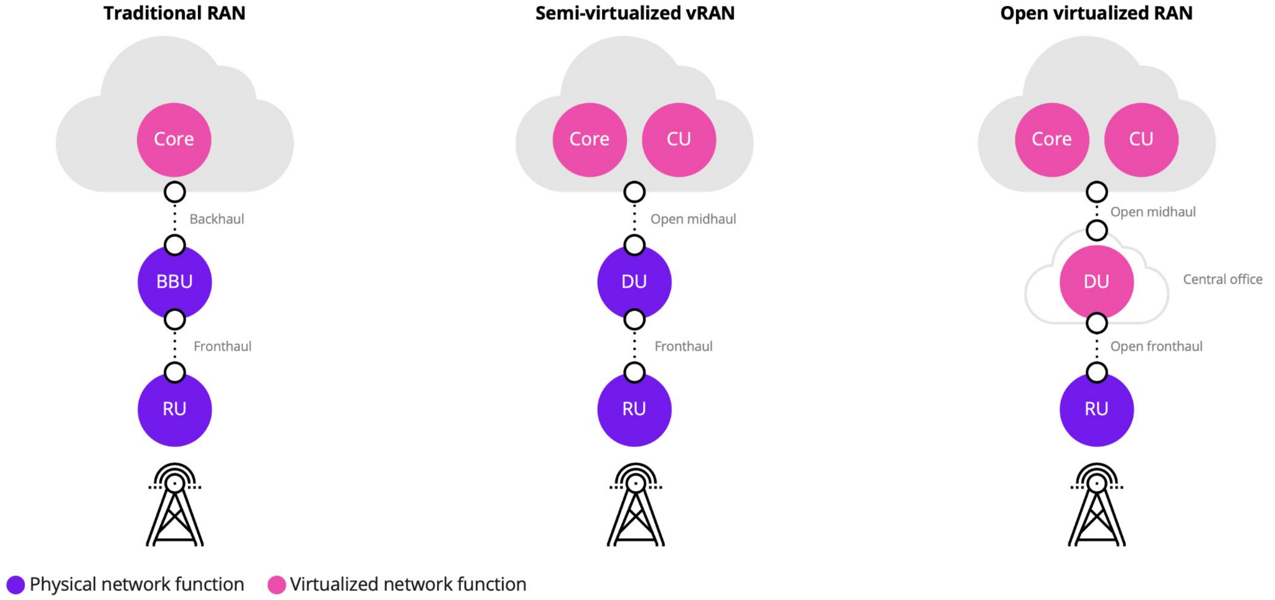
NETWORK FUNCTION	DEFINITION	DESCRIPTION
NEF	Network exposure function	This NF exposes the 5G network's capabilities and services through established application programming interfaces (APIs), facilitating interaction with third parties, such as application developers.
NRF	Network repository function	The NRF provides a record of all NFs available on the platform, together with a profile of each and the services they support. When a new NF is brought up, it registers its IP address and capabilities with the NRF. A consumer NF will request the NRF to identify which NFs are registered with it and have the required capabilities and can act as a producer of services. It interacts with the SCP.
NSSF	Network slice selection function	The NSSF analyzes the requirements of a UE and matches those requirements with a network slice. In today's network slicing, the slices are predefined rather than autonomously orchestrated and for this, this NF must find the best match. The NSSF interacts with AMF, SMF, and PCF.
SCP	Service communications proxy	A critical function to create signaling efficiencies when a consumer NF requests a service from a producer NF. The SCP acts as intermediary between the two, shielding some of the complexities of the infrastructure. The SCP also communicates with the NRF.
SEPP	Security edge protection proxy	A security NF used for roaming and securing the network at the extremities ahead of interfacing with another network. Compared to previous generation networks, 5G networks use enhanced authentication procedures, such as mutual authentication to secure the communication between the device as it roams, and the SEPP plays a role in this with external networks.
CHF	Charging function	The CHF provides online and offline charging features for multiple services. Other measurable qualities beyond throughput will become key with 5G networks, such as latency, slice bandwidth, reliability, availability, security, and APIs, so the CSPs must be able to measure and charge for these.
PCF	Policy control function	The PCF considers device profile, subscription information, and real-time information to enforce rules for traffic steering, QoS, and charging. It also relies on information from other NFs, such as the network exposure function (NEF) and SMF.

5G Core Service-Based Architecture (SBA)

NETWORK FUNCTION	DEFINITION	DESCRIPTION
NWDAF	Network data analytics function	This function provides data analytics of the operations and services of the network by processing and analyzing the network performance, user traffic patterns, and network load to deliver actionable intelligence to optimize QoS and overall user experience.
NSSMF	Network slice subnet management function	This function takes the slice requirements from the NSMF and ensures the necessary NF resources are made available and are instantiated.
NSMF	Network slice management function	This NF will keep the necessary information for several popular network slice templates for known use cases for fast provisioning.
MDAF	Management and data analytics function	The MDAF provides analytics information from network slices to ensure optimization of resources.

RAN Evolution to Open vRAN

FIGURE 16
RAN evolution to open vRAN

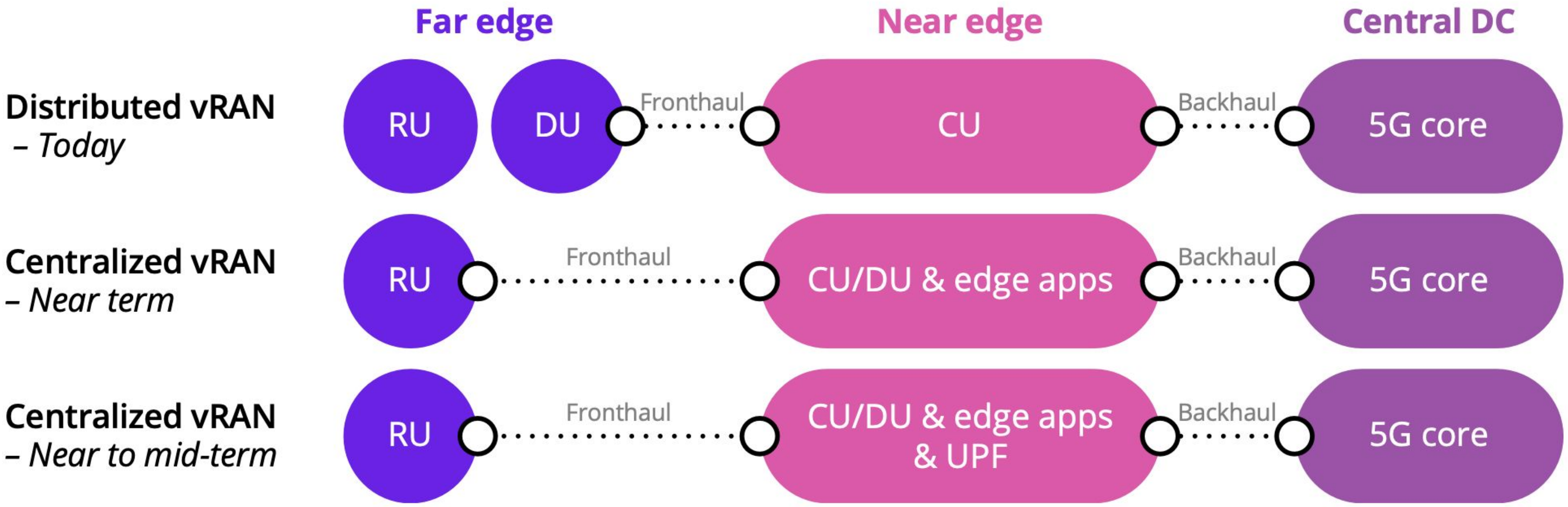


SOURCE: OMDIA

RAN Evolution to Open vRAN

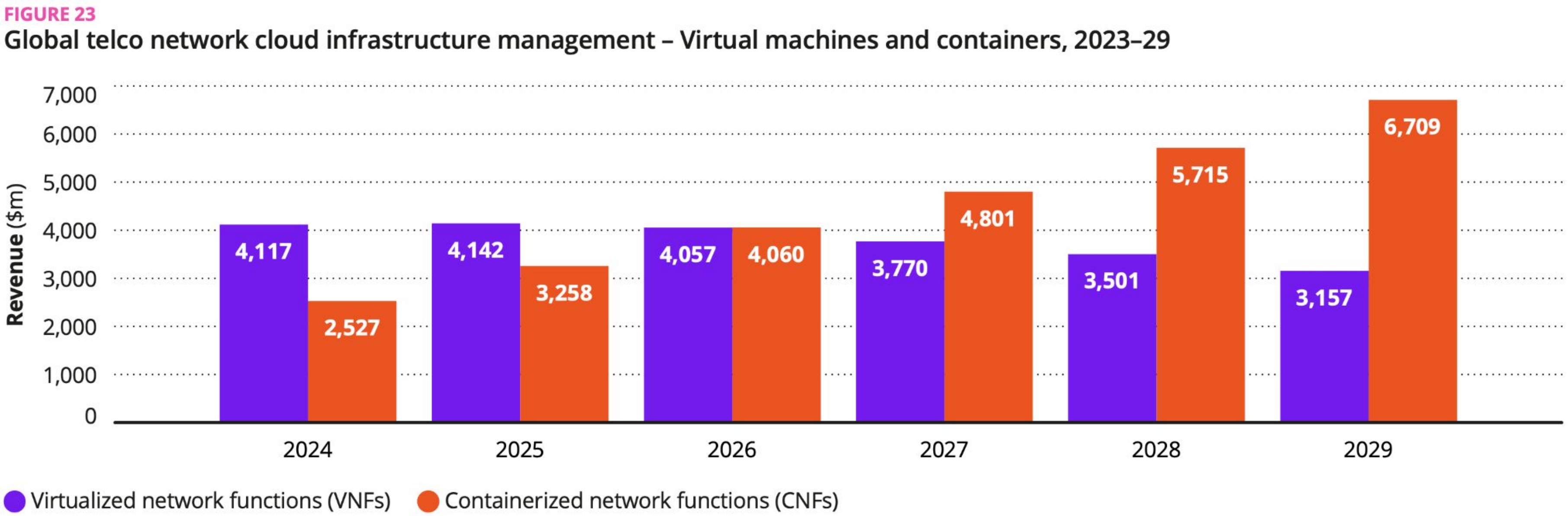
FIGURE 17

Network architecture will evolve to support new use cases



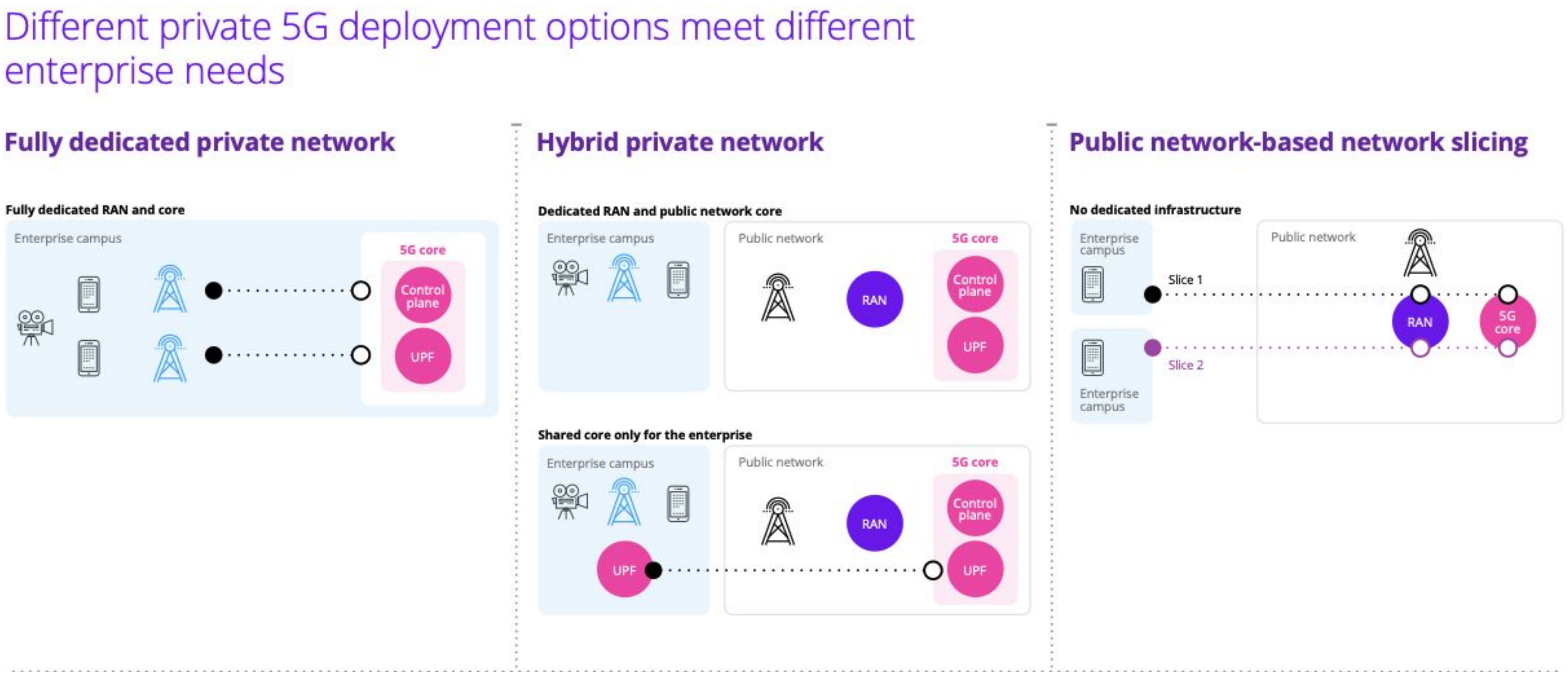
SOURCE: OMDIA

Virtual Machines (VMs) vs Container Network Functions (Projection)

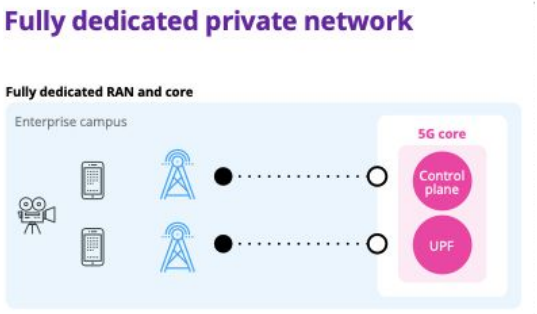


SOURCE: OMDIA TELCO NETWORK CLOUD TRACKER – 2024 ANNUAL FORECAST REPORT **NOTE:** N=135

Private 5G deployment options for enterprise



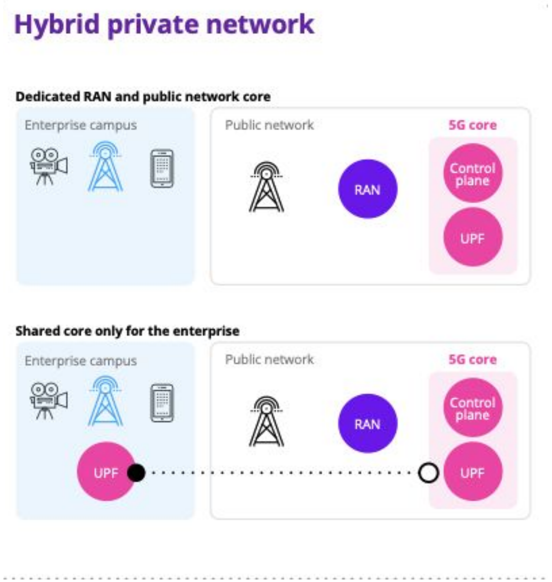
Private 5G deployment options for enterprise



- All the infrastructure (RAN and core) is deployed for the sole use of the enterprise
- The enterprise is in full control of the network
- Data remain within the enterprise control, no public network
- Network can be tailored to all the specific needs
- ideal for enterprises with critical needs such as mines and factories

Private 5G deployment options for enterprise

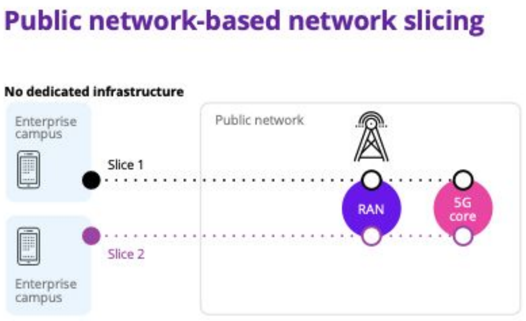
- Infrastructure shared between the private and the public network
- Provides the greatest flexibility for the provider and for the enterprise



- Dedicated RAN, shared core from the public network
- Part of the core can be deployed on site for the enterprise

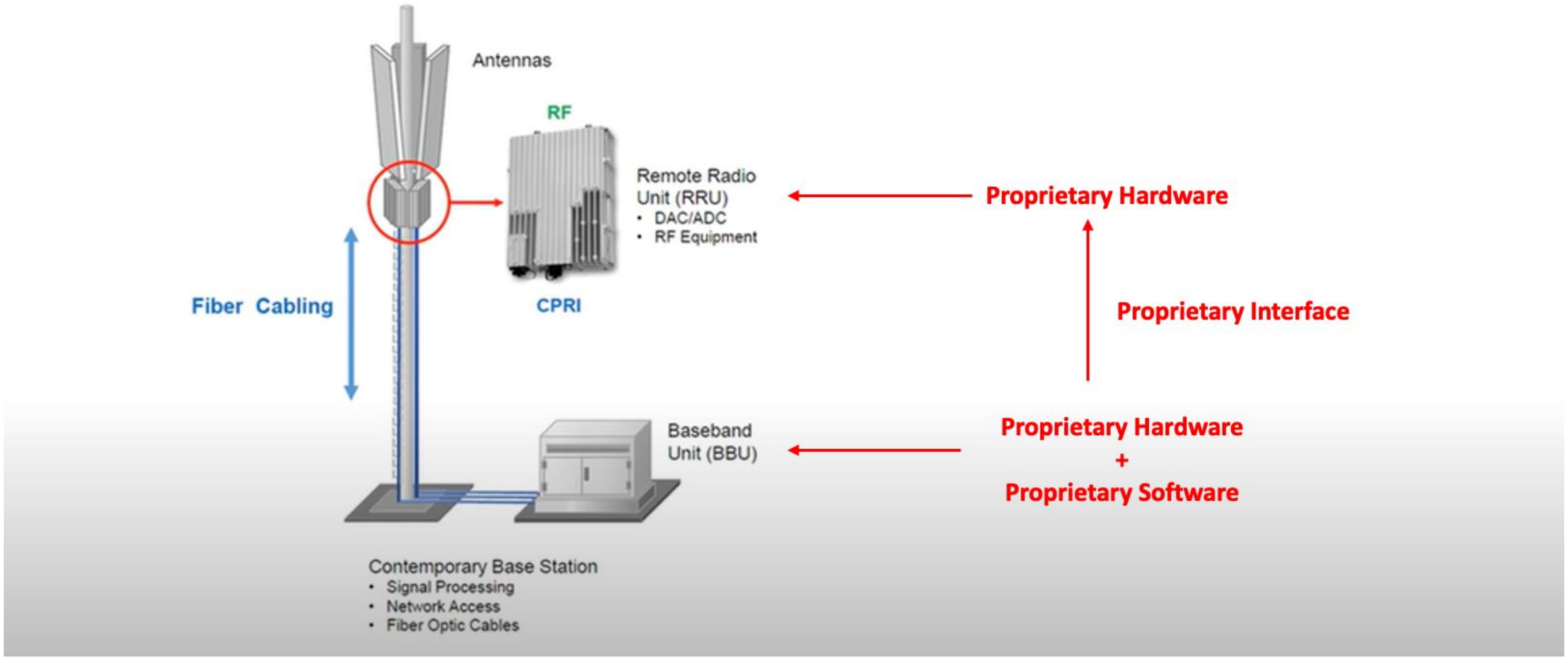
Private 5G deployment options for enterprise

- The infrastructure belongs to the public network
- The fastest option. No additional infrastructure needs to be deployed by the enterprise
- The infrastructure capex is already borne by the telecom operator rather than the enterprise
- Network slicing can support temporary networks which may require to be active only for limited amount of time (or limited time during the year)
- Ideal for enterprises with very large coverage needs or with limited budgets



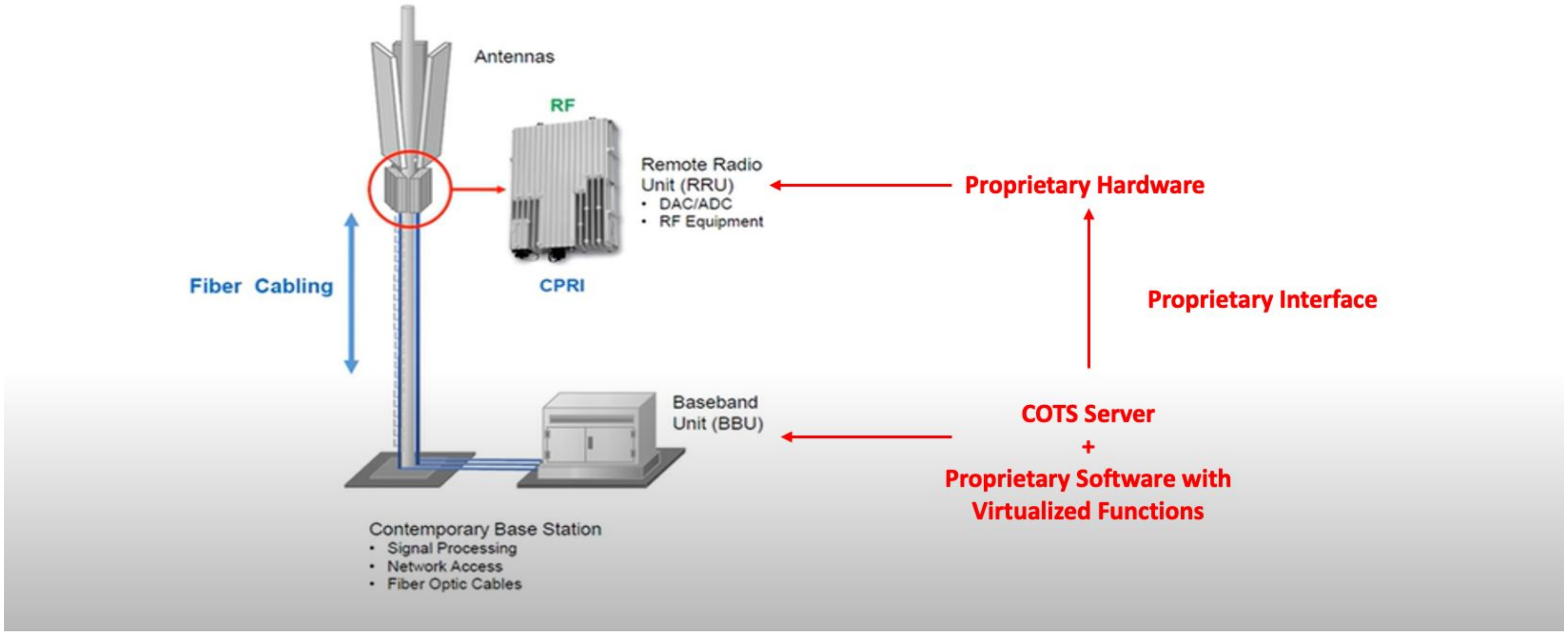
O-RAN

Legacy RAN Deployments



Virtualized RAN Approach

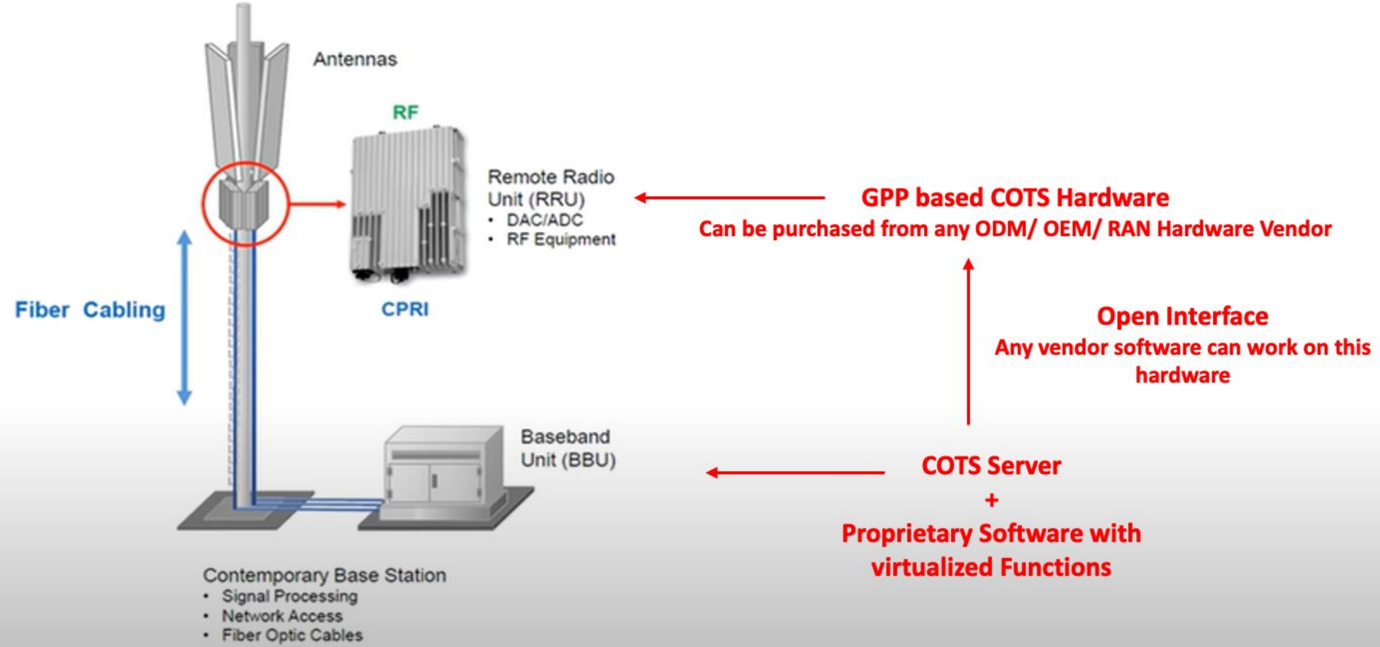
COTS = Commercial Off the Shelf



vRAN is not necessarily Open RAN

O-RAN

ODM (Original Design Manufacturer)
OEM (Original Equipment Manufacturer)
GPP (General Purpose Processor)



O-RAN Vs Open vRAN

Characteristic	O-RAN	Open vRAN
Origin	Defined by the **O-RAN Alliance**	General concept; focuses on vRAN
Goal	Open and disaggregated RAN architecture	Virtualization of RAN (vRAN)
Interfaces	Open and standardized interfaces	Not necessarily open interfaces
Architecture	Disaggregated (CU/DU/RU + RIC)	Virtualized on COTS hardware
Flexibility	Promotes multi-vendor interoperability	Focuses on hardware-software decoupling
Key Technologies	RIC, open interfaces (e.g., fronthaul)	NFV, VNFs, containers, COTS servers

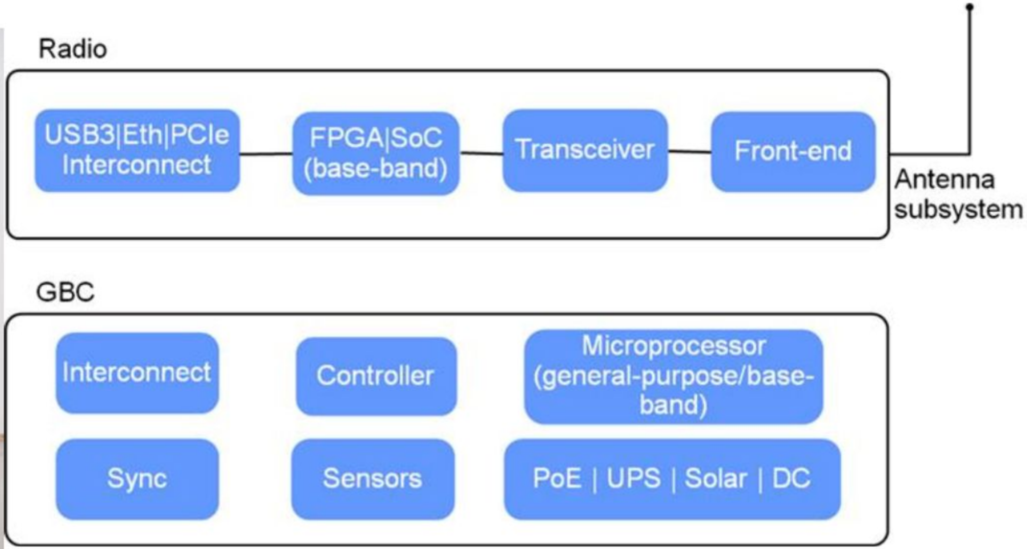
GPP Vs SPP

- A general purpose processor (GPP) is generally called a Central Processing Unit (CPU). It performs all of the general work of the computer.
- Intel x86, ARM, MIPS, SPARC, RISC-V are examples of GPP
- True hardware / software disaggregation possible
- The main advantage of GPP is that:
- Due to large volumes, economy of scale is achievable
- Costs are lower in high volumes
- Faster pace of innovation due to software-driven development.

- A single purpose processor (SPP) has a limited number of functions and would be optimized for a specific area.
- An example might be a MAC processor from DSP, you might build an FPGA or ASIC using several of these to create a digital filter for some specific application.
- It would run much faster than running the same algorithm on a general CPU, probably would use a lot less power.

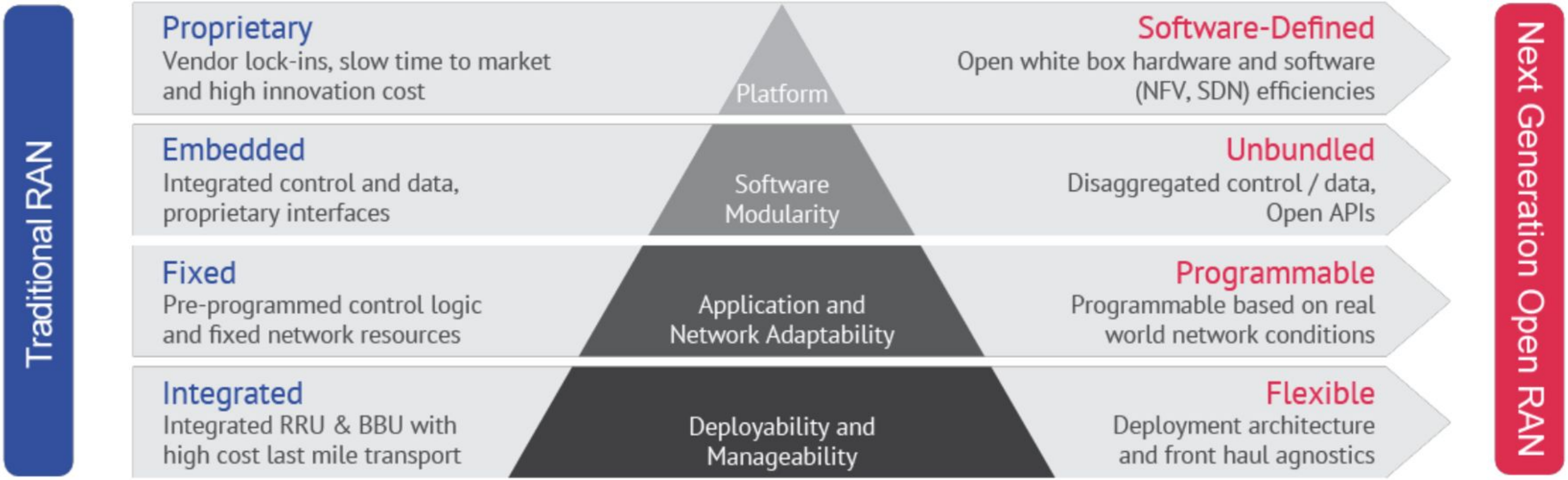
Open RAN: Enabling White Box RAN

- Whitebox RAN = All-in-one GPP based base station
- But you can have a Whitebox RRU + COTS BBU



General-purpose and Base-band Computing (GBC)

Traditional RAN vs Open RAN

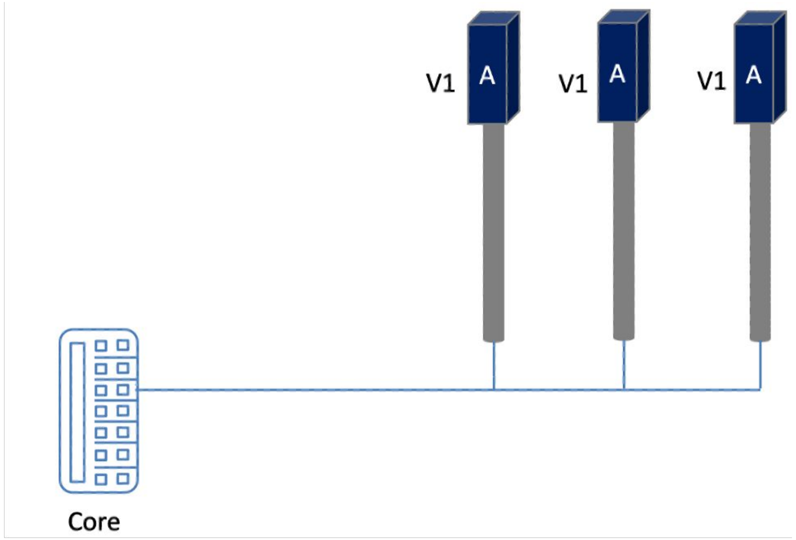


Source: https://www.slideshare.net/Radisys/getting-to-the-edge-exploring-4g5g-cloudran-deployable-solutions

So why is Open RAN so interesting?

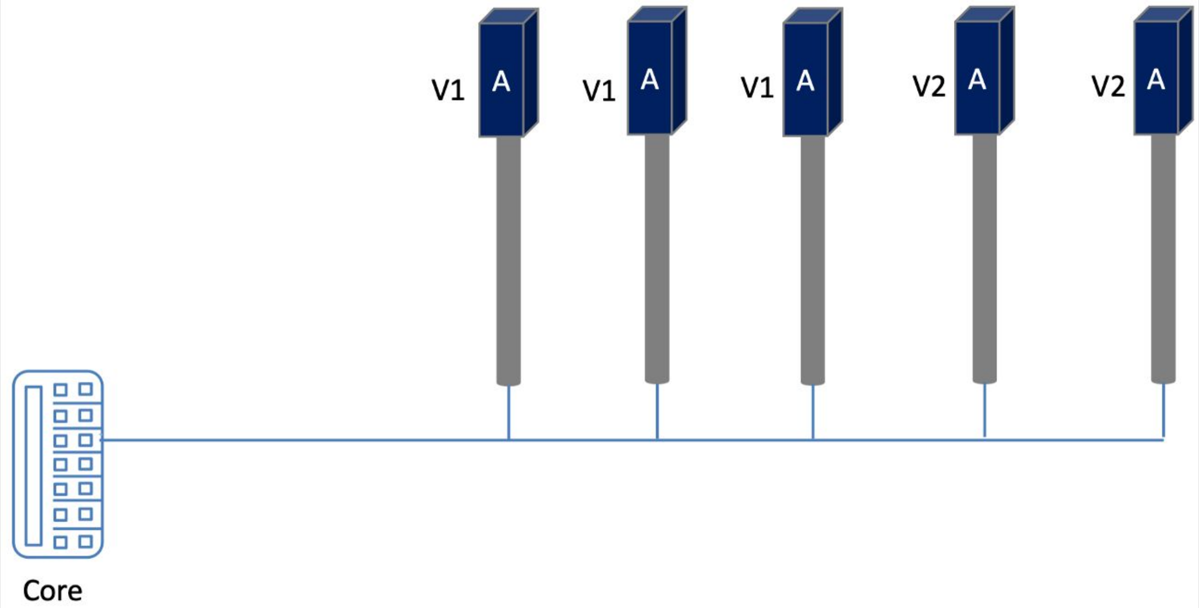
Lets take an example scenario of an operator (MNO) or service provider (SP)

It rely only on vendor V1



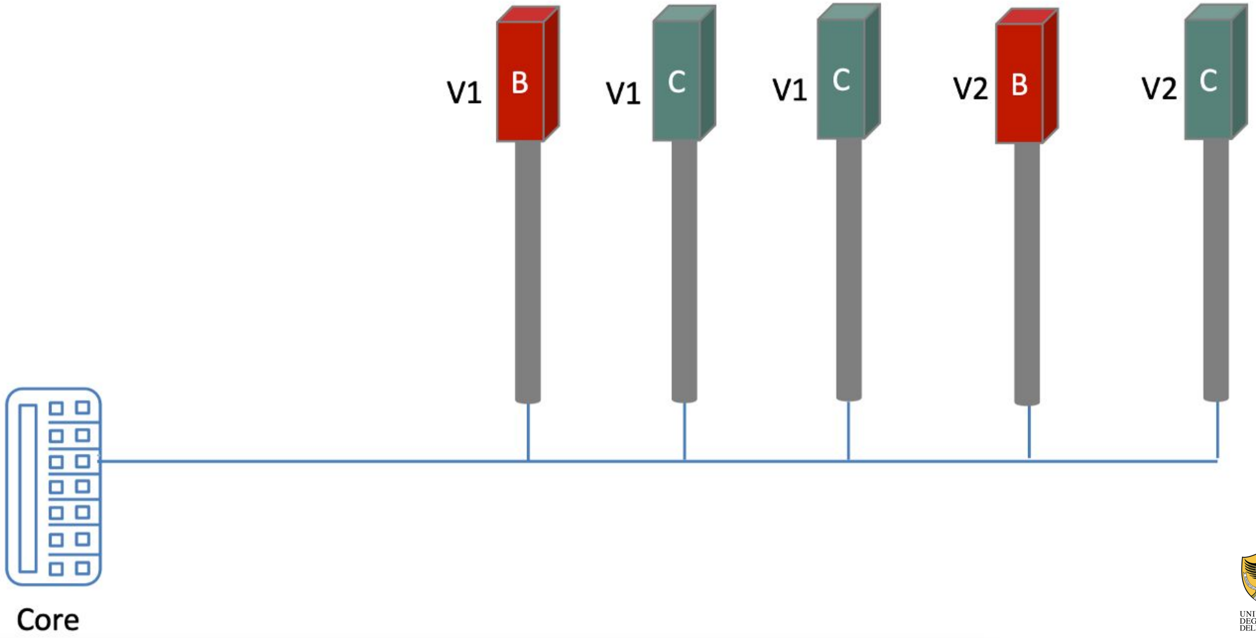
So why is Open RAN so interesting?

When the MNO is looking to expand the network, it may decide to introduce vendor V2



So why is Open RAN so interesting?

If the hardware from vendor 1 is not performing well, the MNO may start replacing the RAN while keeping same software vendor



The O-RAN Alliance

O-RAN ALLIANCE is a world-wide community of mobile operators, vendors, and research & academic institutions with the mission to re-shape Radio Access Networks to be more intelligent, open, virtualized and fully interoperable.

O-RAN Ecosystem

Founded in Feb 2018, it now has 33 MNOs as members including JIO & more than 260 RAN solution providers as contributors

ORAN is about Operator Defined Next Generation RAN Architecture and Interface with key objective of

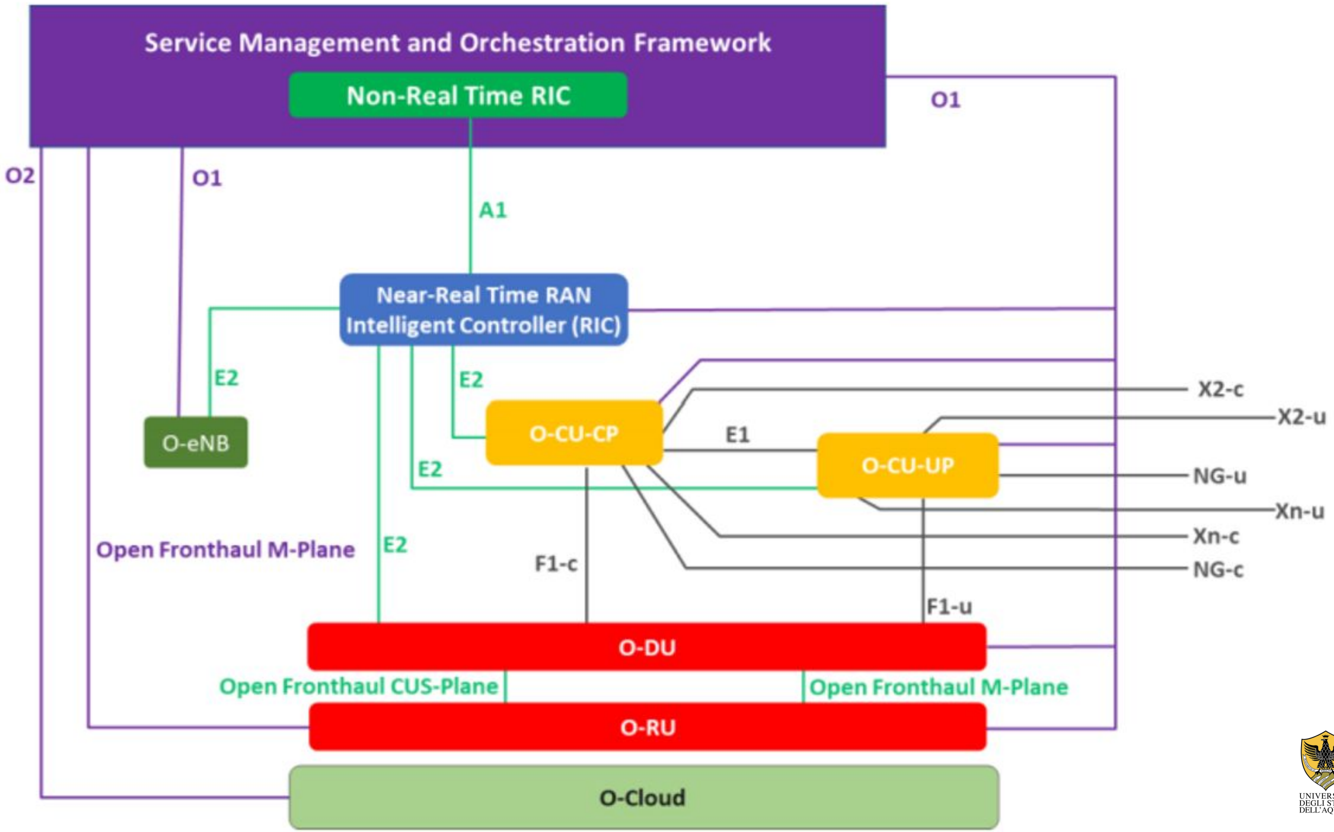
- Transforming Radio Access Networks open. Intelligent, virtualized and fully interoperable RAN
- Providing flexibility to the MNOs to choose the best breed of RAN Components

Operator Members

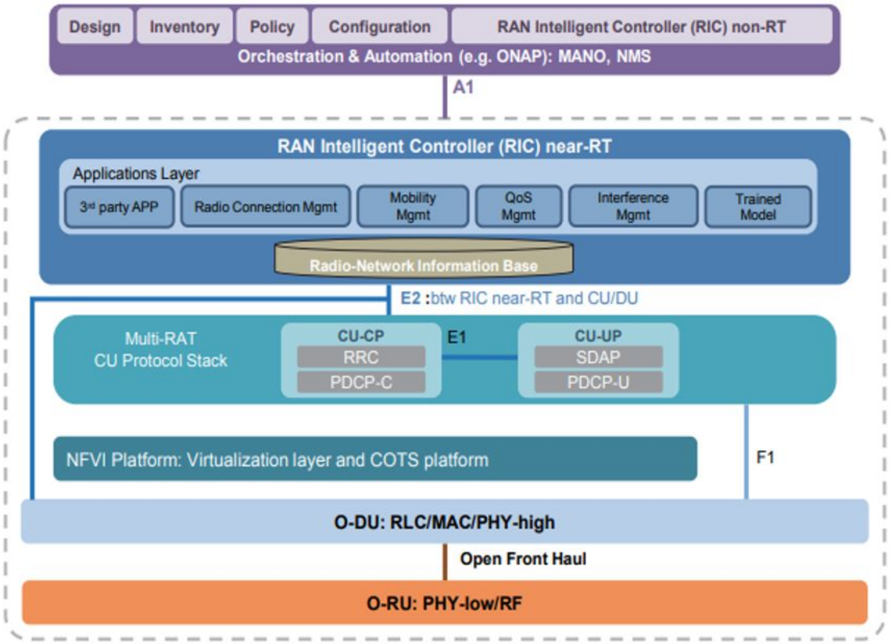
Benefits of O-RAN Architecture

- Multi-Vendor deployment
- Capable of reducing network CAPEX & OPEX
- Capable of improving network efficiency and performance
- Increases scalability
- Endless innovation and healthy competition among vendors aids overall development

O-RAN Overall Architecture



O-RAN Overall Architecture

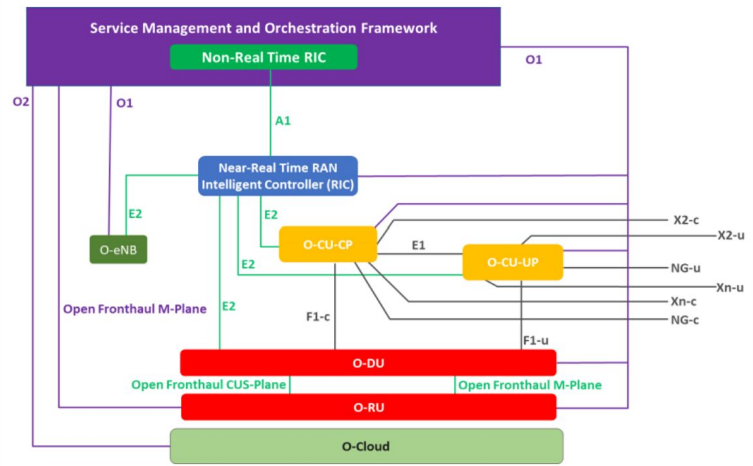


Source: O-RAN Alliance

CP: Control Plane
MAC: Media Access Control
MANO: Management and Orchestration
NFVI: Network Functions Virtualisation Infrastructure
ONAP: Open Network Automation Platform
PDCP: Packet Data Convergence Protocol

PHY: PHYsical layer
RAT: Radio Access Technology
RF: Radio Frequency
RLC: Radio Link Control
RRC: Radio Resource Control
SDAP: Service Data Adaptation Protocol
UP: User Data Plane

O-RAN Overall Architecture

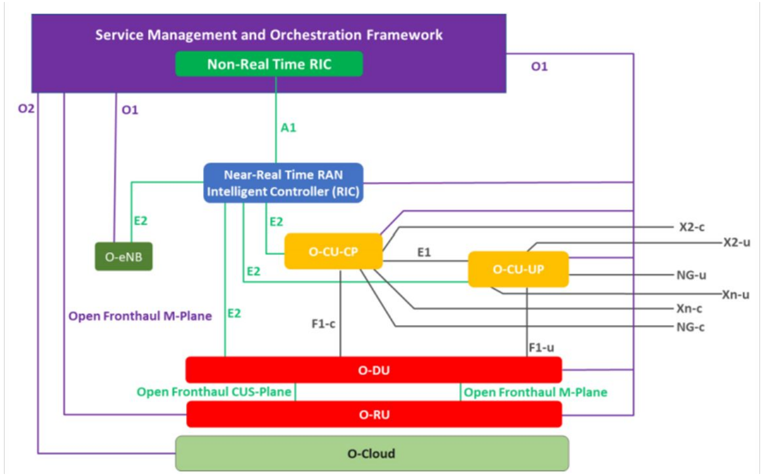


Service Management and Orchestration (SMO)

- This framework contains the Non-RT RIC function (>1s) which has the goal of supporting intelligent RAN optimization in non-real-time (> 1s) by providing policy-based guidance using data analytics and AI/ML training/inference.

Deployed as VNFs or containers to distribute capacity across multiple network elements with security isolation and scalable resource allocation.

O-RAN Overall Architecture

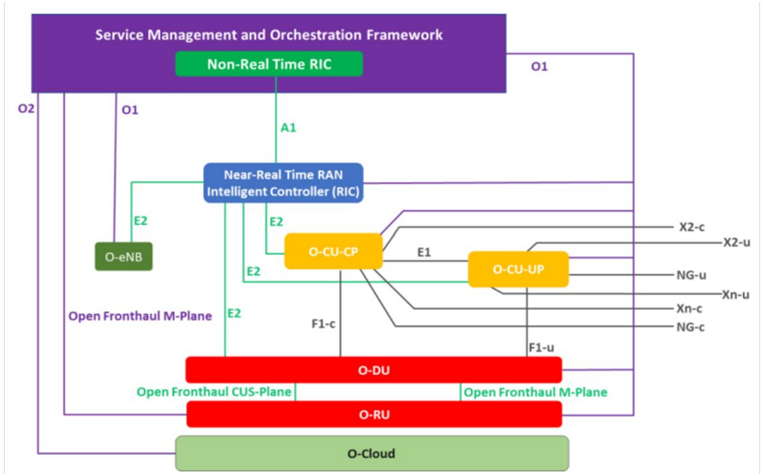


<u>Near-RT RIC</u> (< 1s)

- Near-RT RIC, O-CU-CP, O-CU-UP, O-DU, and O-RU are the network functions for the radio access side
- Near-RT RIC enables near real-time control and optimization of O-RAN (O-CU and O-DU) nodes and resources over the E2 interface with near real-time control loops (i.e., 10ms to 1s)
- The Near-RT RIC hosts xApps that use E2 interface to collect near real-time RAN information to provide value added services using these primitives, guided by the policies and the enrichment data provided by the A1 interface from the Non-RT RIC

Deployed as VNFs or containers to distribute capacity across multiple network elements with security isolation and scalable resource allocation.

O-RAN Overall Architecture

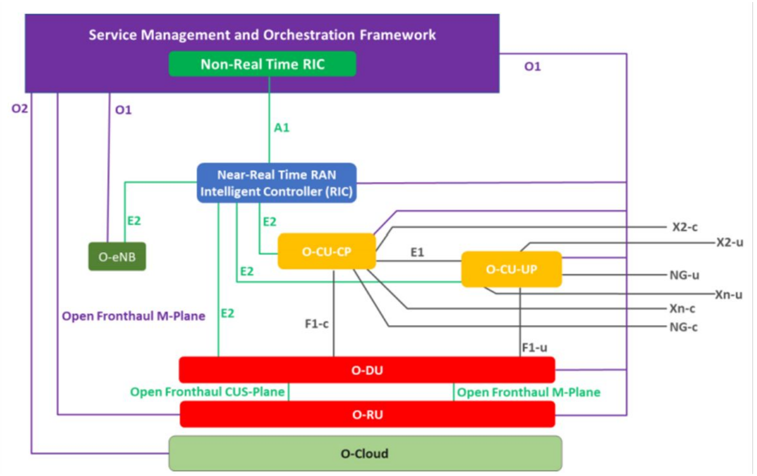


Multi-RAT CU Protocol Stack

- This function layer supports various protocol stacks including 4G and 5G Radio access
- Supports virtualization and consists of function that executes commands issued by non-RT RIC
- Supports F1/E1/E2/X2/Xn interfaces specified by 3GPP

Deployed as VNFs or containers to distribute capacity across multiple network elements with security isolation and scalable resource allocation.

O-RAN Overall Architecture

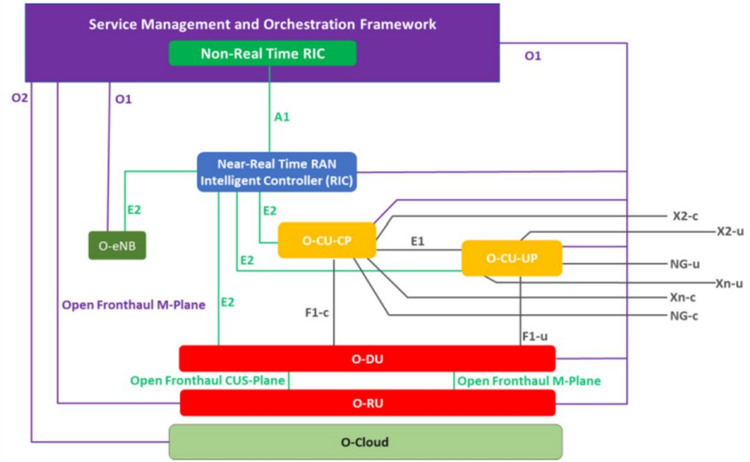


<u>O-DU and O-RAN Radio Unit(O-RU)</u>

- O-DU and O-RU consists of real time L2 functions and functional group performing Baseband processing and radio signal processing
- Open Fronthaul are newly specified between O-DU and O-RU

Only O-DU may be deployed as VNF or container to distribute capacity across multiple network elements with security isolation and scalable resource allocation.

O-RAN Overall Architecture

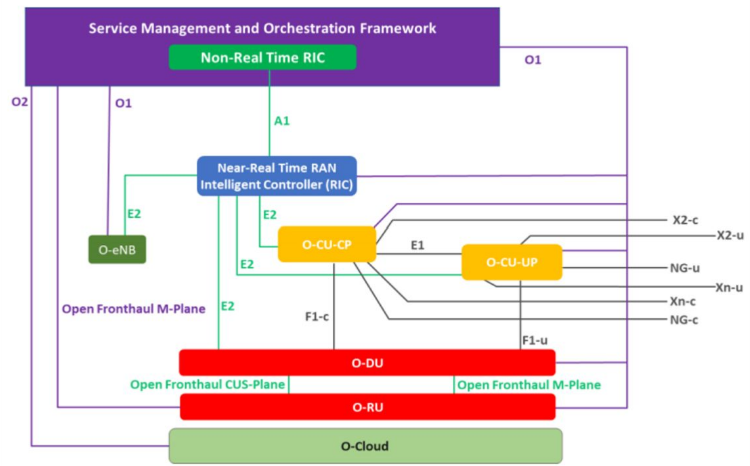


<u>O-Cloud</u>

- Virtual Infrastructure Management Framework
- Infrastructure – COTS / White Box / Peripheral Hardware and Virtualization Layer

Deployed as VNFs or containers to distribute capacity across multiple network elements with security isolation and scalable resource allocation.

O-RAN Overall Architecture



Interfaces:

A1 is defined between NRT RIC and nRT RIC, through which NRT RIC provides nRT RIC with policies, enrichment info, and ML model updates, while from the other hand, nRT RIC provides back the policy feedback (i.e. how the policy set by NRT RIC works)

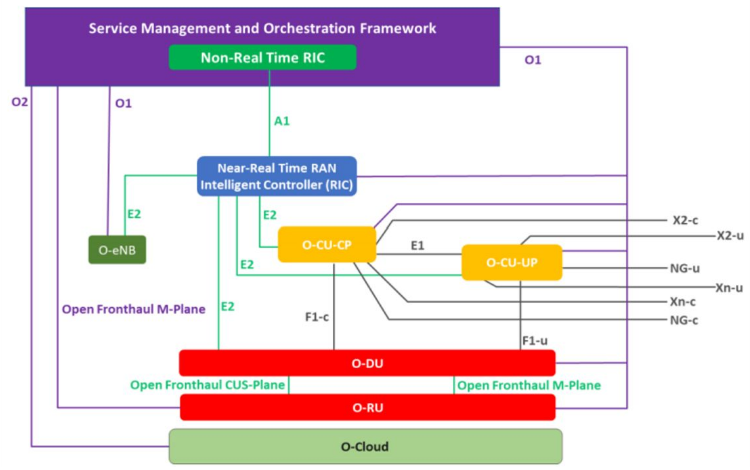
E2 actually touches and gets into the specific entities within the base station, i.e. O-RU, O-DU, O-CU. Therefore, e.g. from one side we can control what is happening within that BS, using monitor, suspend, override, control messages, and execute actions coming from by xApps/nRT RIC, and gets data collection and feedback from those entities.

E1 between CU-CP and CU-UP

F1 between O-CU and O-DU

Open Fronthaul between O-DU and O-RU

O-RAN Overall Architecture



Interfaces:

O1 supports typical FCAPS and other management functions between the ORAN components and the SMO Framework. Interface with configuration, reconfiguration, registration, security, performance, monitoring aspects exchange with individual nodes, like O-CU-UP, O-CU-CP, O-DU, O-RU, as well as nRT RIC. FCAPS:

Fault management: detecting failures and isolating failed component

Configuration management: managing orderly network changes e.g. equipment addition/removal

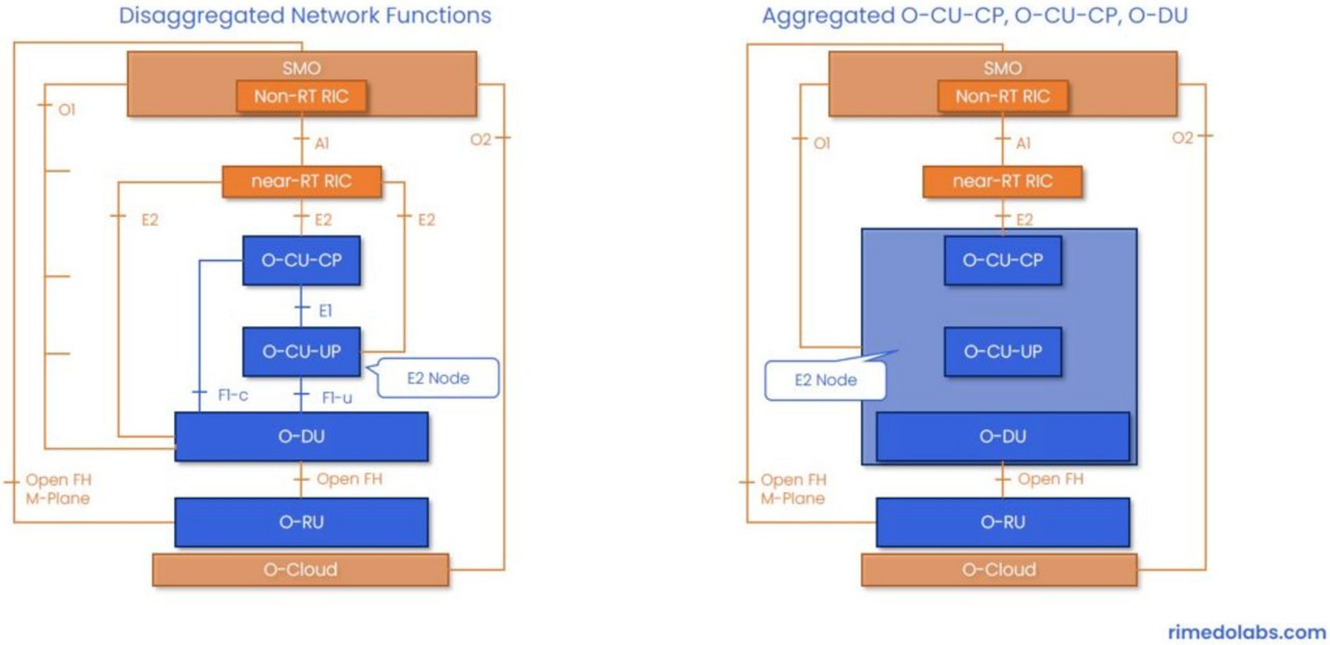
Accounting management: billing and developing component lifetime histories

Performance management: monitoring and managing various network performance metrics

Security management: user authentication, control access to network elements, user data protection etc.

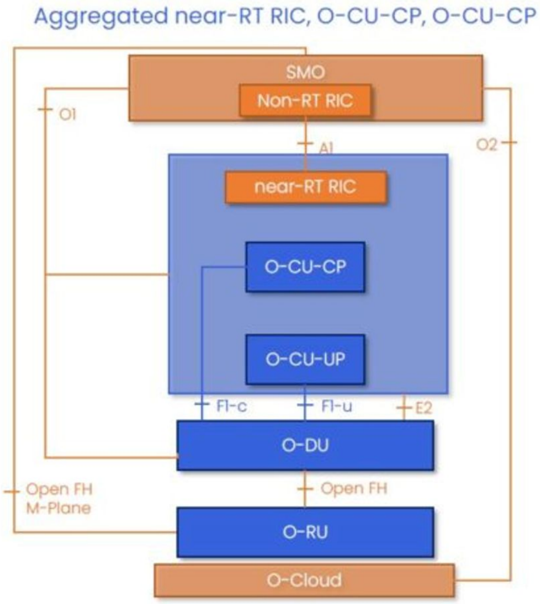
O2 supports virtual resource management functions and other O-Cloud related management functions

O-RAN Implementation Options (1)

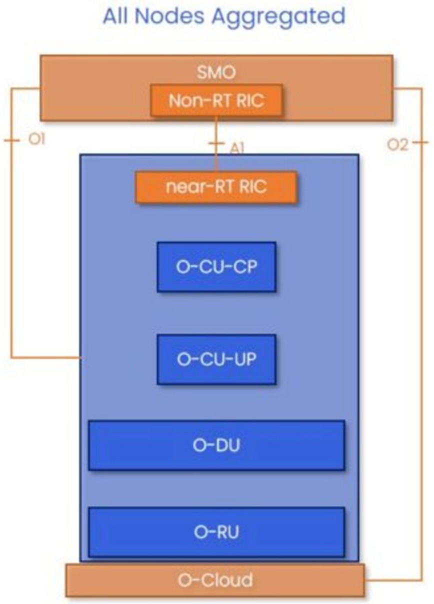


E2 Node is a logical node terminating E2 interface, for NR it is O-CU-CP, O-CU-UP, O-DU or any combination as allowed by O-RAN Alliance.

O-RAN Implementation Options (2)



nRT RIC incorporated with O-CUs, which means that the E2 interface to control O-CUs is internal, and from that combined node there is a regular E2 interface towards O-DU only.



rimedolabs.com

All the nodes (except SMO) are combined together, thus the E2 interface is fully internal, and there is only a single O1 connection, and the always present A1 interface.

RAN Energy Consumption: State of the Art

Theoretical Formulations and Heuristic Methods

- Focus
  - Uses various metrics
    - **Carbon Usage Effectiveness (CUE)**
    - **Water Usage Effectiveness (WUE)**
- Techniques
  - Integer Linear Programming (ILP)
  - Mixed Integer Linear Programming (MILP)

optimal resource allocation solutions tailored to specific objectives

- Challenges
  - Limited practicality due to long computation times.
  - Often impractical for real-time or large-scale systems.

Hardware and Software-Based Power Monitoring

- Granular Insights via Hardware
  - Advanced tools ⇒ Precise power measurement at the component level.
- Software Solutions
  - Algorithms and tools ⇒ Power profiling and energy footprint measurement of applications.
- Data Center-Wide (DC) Monitoring Systems
  - Dynamo ⇒ Provides comprehensive power management across entire data centers.
- Challenges
  - Potential for sub-optimal solutions.
  - Limitations in real-time monitoring or coverage at specific stack levels.

Importance of Energy Monitoring

Figure 15

Top reasons for measuring energy efficiency

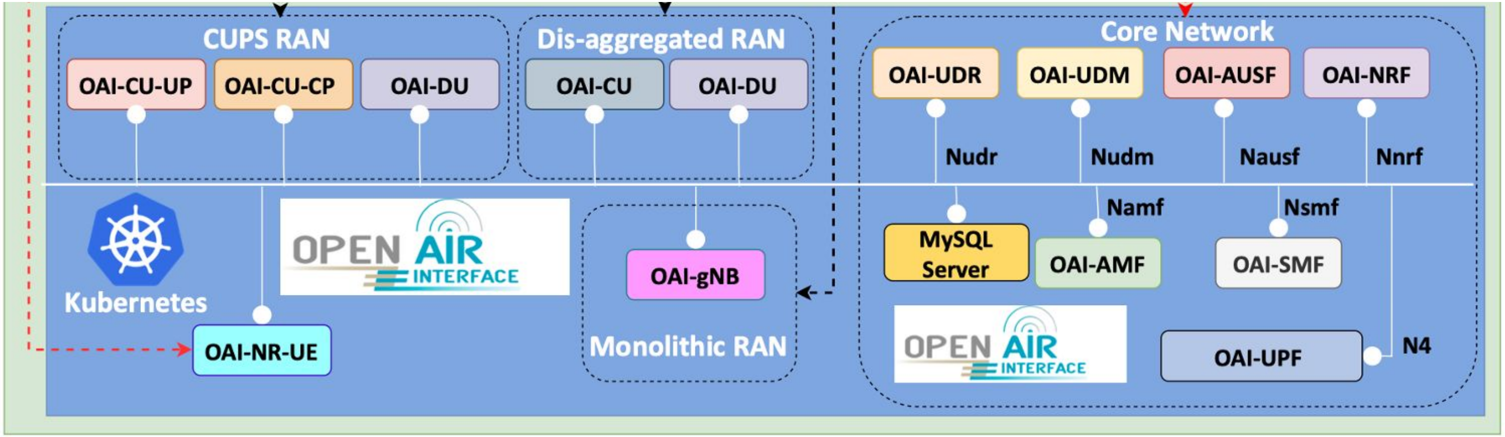
How important are the following factors in measuring energy-efficiency indicators for your company?

More important

1. Track our company's progress over time (including targets)
2. Better understand and manage energy costs
3. To comply with reporting requirements/regulations
4. To include them in voluntary reporting and disclosures
5. Compare networks within our company
6. Compare with other operators/networks
7. Other (energy mix, finding optimum capex/opex relation)

Source: GSMA Intelligence Workshop Operator Survey 2024

Our Developed OAI deployments



Introduction to Virtualization

and Containers

What is Docker?

docker®

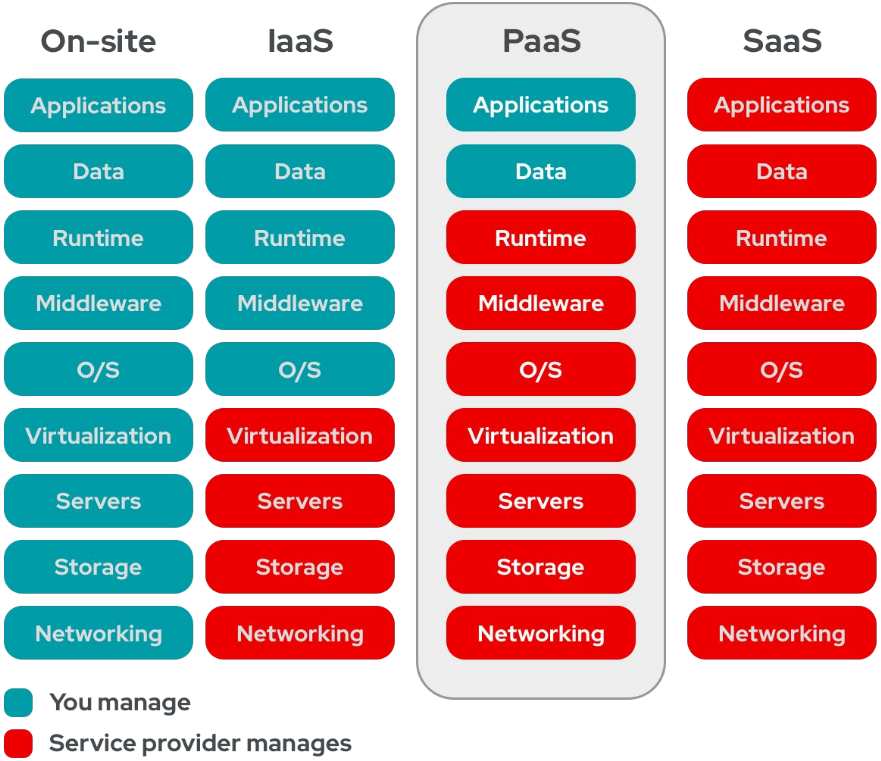
Intro

Docker is a set of platform as a service (PaaS) products that uses OS-level virtualization to deliver software in packages called containers.

Intro

*Docker is a set of **platform as a service** (PaaS) products that uses OS-level virtualization to deliver software in packages called containers.*

Platform as a Service (PaaS)



Platform as a Service (PaaS)

*Docker is a set of platform as a service (PaaS) products that uses **OS-level virtualization** to deliver software in packages called containers.*

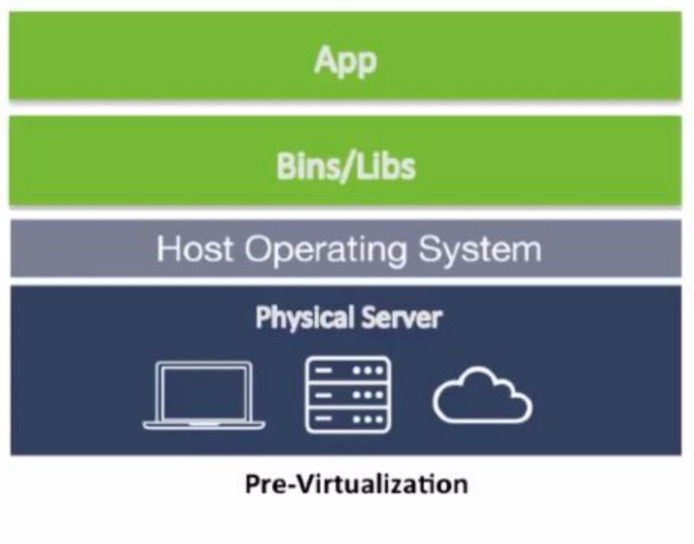
pre-Virtualization - Bare metal OS

Pro:

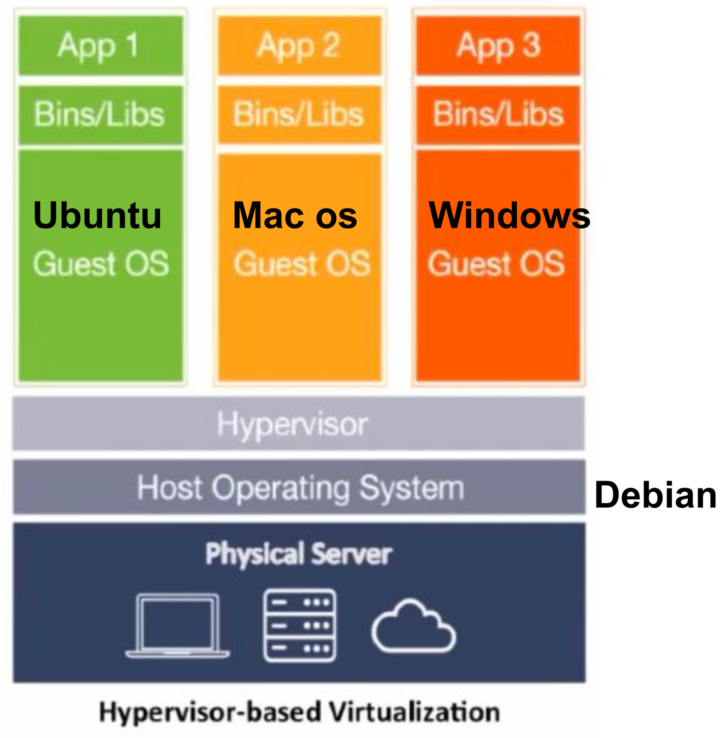
- The most performant solution

Con:

- Huge cost
- Slow Deployment
- Hard to Migrate



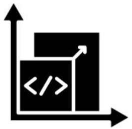
Hypervisor-based Virtualization



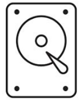
Hypervisor providers:

- VMware, Virtual Box, ...

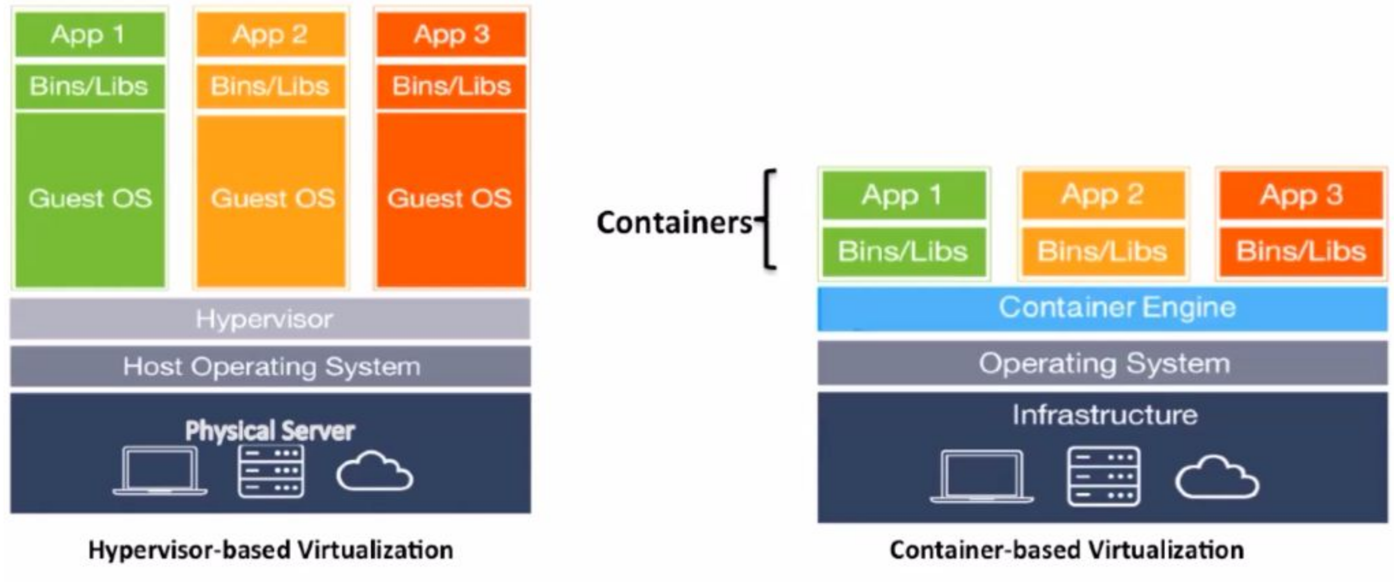
Pro:

- Cost-Efficient 
- Easy to scale (Vertically) 

Con:

- Kernel Resource Duplication
- Application Portability Issue
- Huge disk resources 

OS-based Virtualization (Container-based Virtualization)



Pro:

- Cost-Efficient
- Guaranteed portability
- Resource optimization?

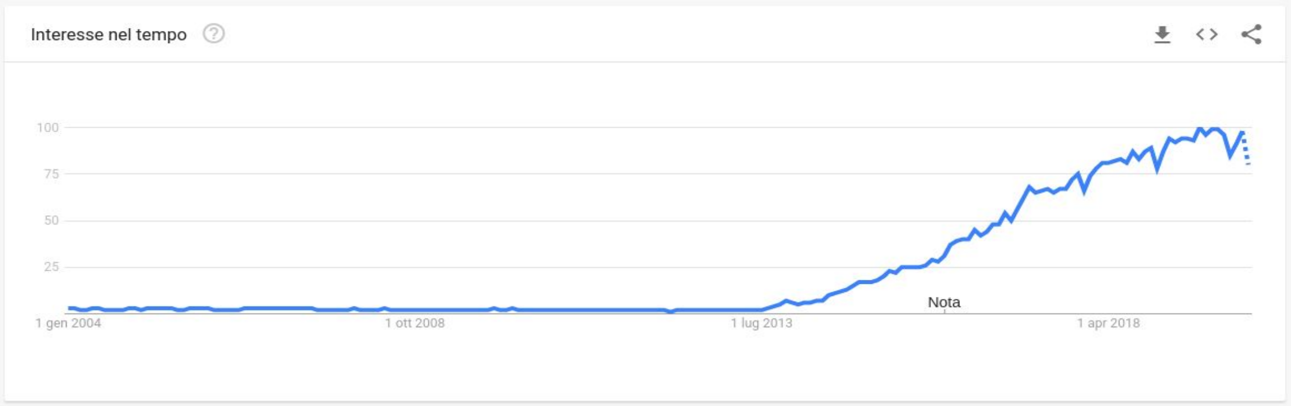
- Fast Deployment
- Runtime Isolation
- Easy to scale (Horizontally)

Platform as a Service (PaaS)

*Docker is a set of platform as a service (PaaS) products that uses OS-level virtualization to deliver software in packages called **containers**.*

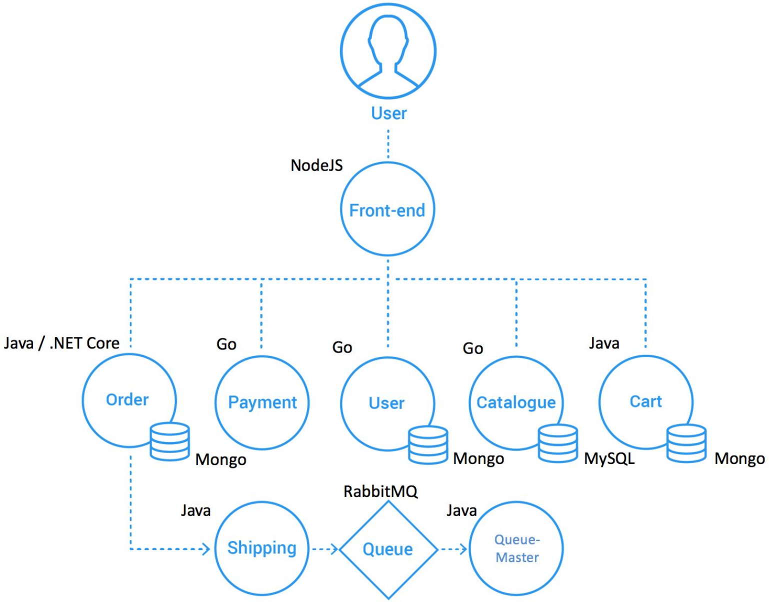
Containers

- Containers offer a **logical packaging** and **isolation** mechanism
- Applications can be abstracted from the environment in which they actually run
- Packaging allows container-based applications to be **deployed** easily and **consistently**
- Developers are able to create predictable **environments** that are isolated from the rest of the applications and can be run anywhere
- Give more granular control over resources



An architectural mindset

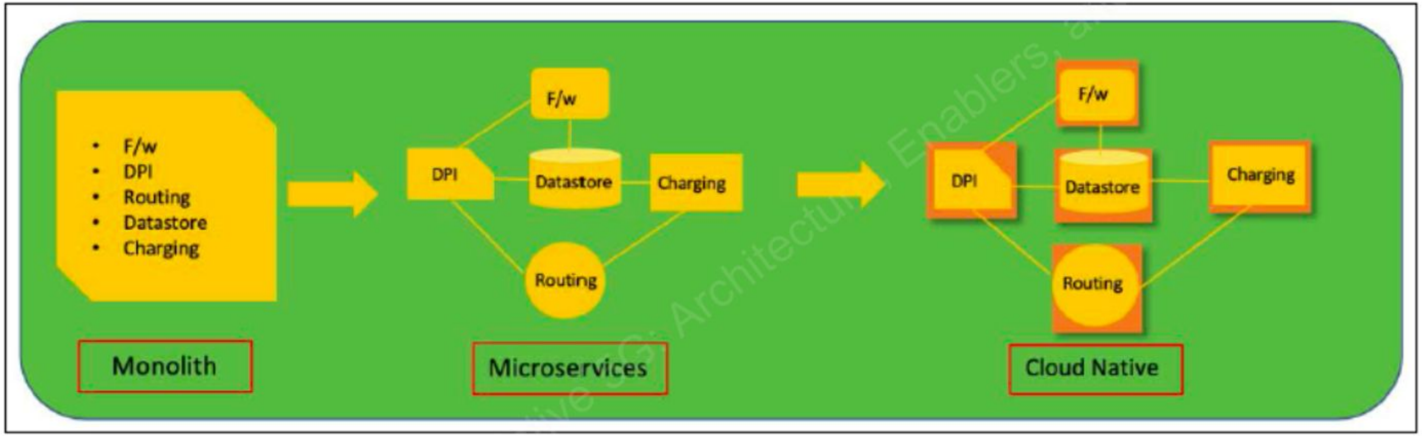
Containers are designed to support the Microservice Architectures



An architectural mindset

Containers support Microservice-based Architectures

Cloud-native Architectures are aware of the container world and the cloud



(5G Americas)

The CCNF

https://www.cncf.io/

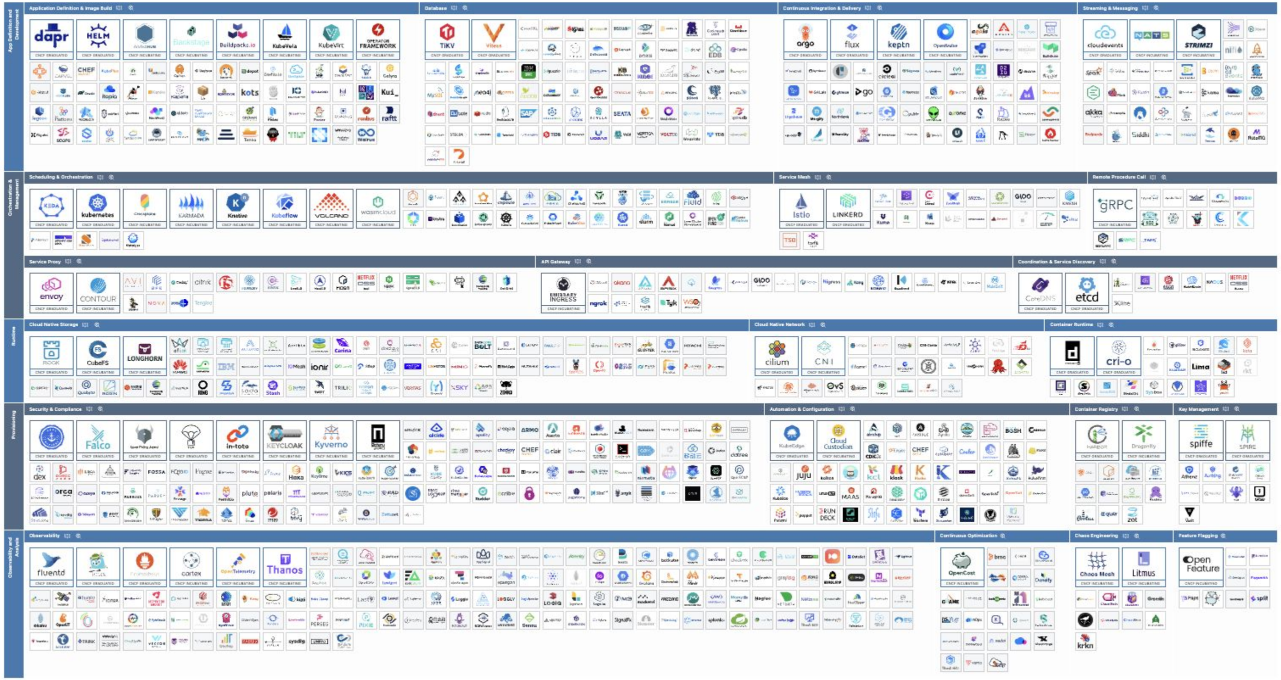
The **Cloud Native Computing Foundation** (CNCF):

- Is an open-source foundation that supports and promotes cloud-native technologies
- Was founded in 2015 as part of the Linux Foundation
- **Mission**: To drive adoption of cloud-native technologies by fostering an open, collaborative ecosystem

Cloud-Native: Systems built for dynamic, scalable, and resilient cloud environments

The CCNF Landscape

https://landscape.cncf.io/



Reasoning questions

What consumes energy?

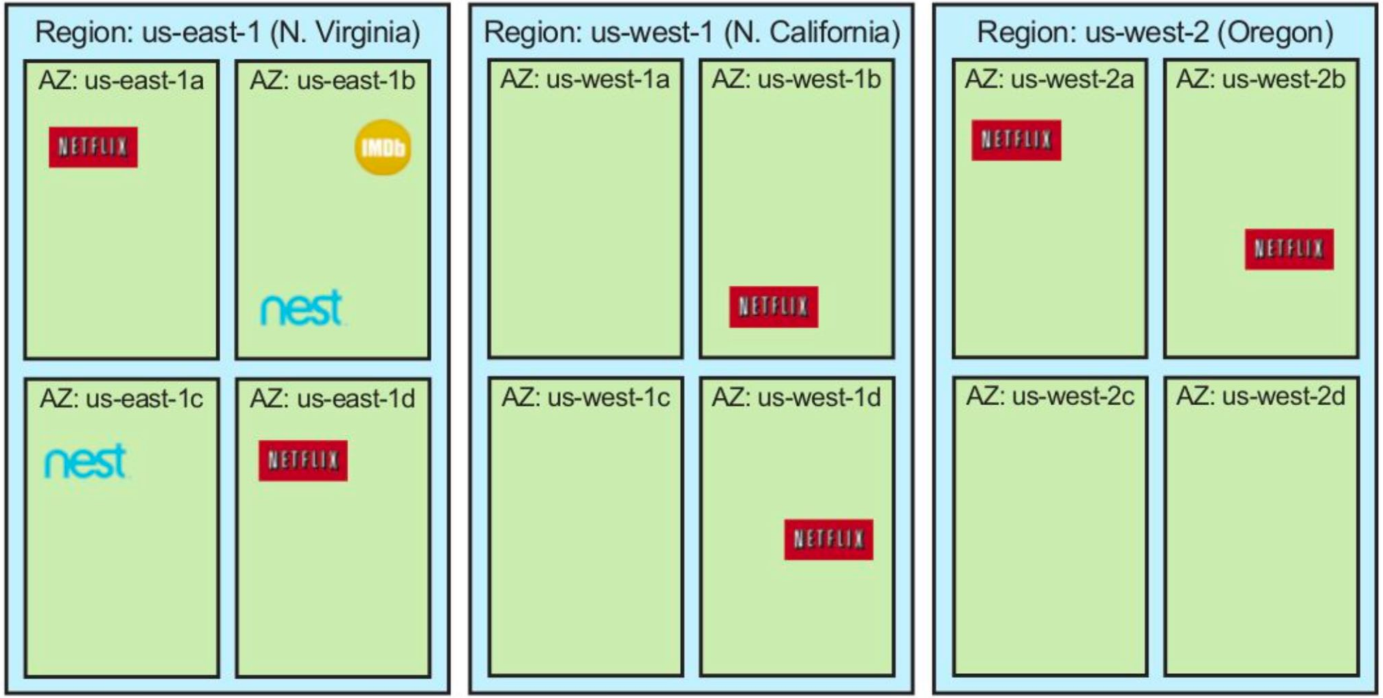
1. The Applications
2. The Containers
3. The Operating System
4. The Hardware

Reasoning questions

What consumes energy?

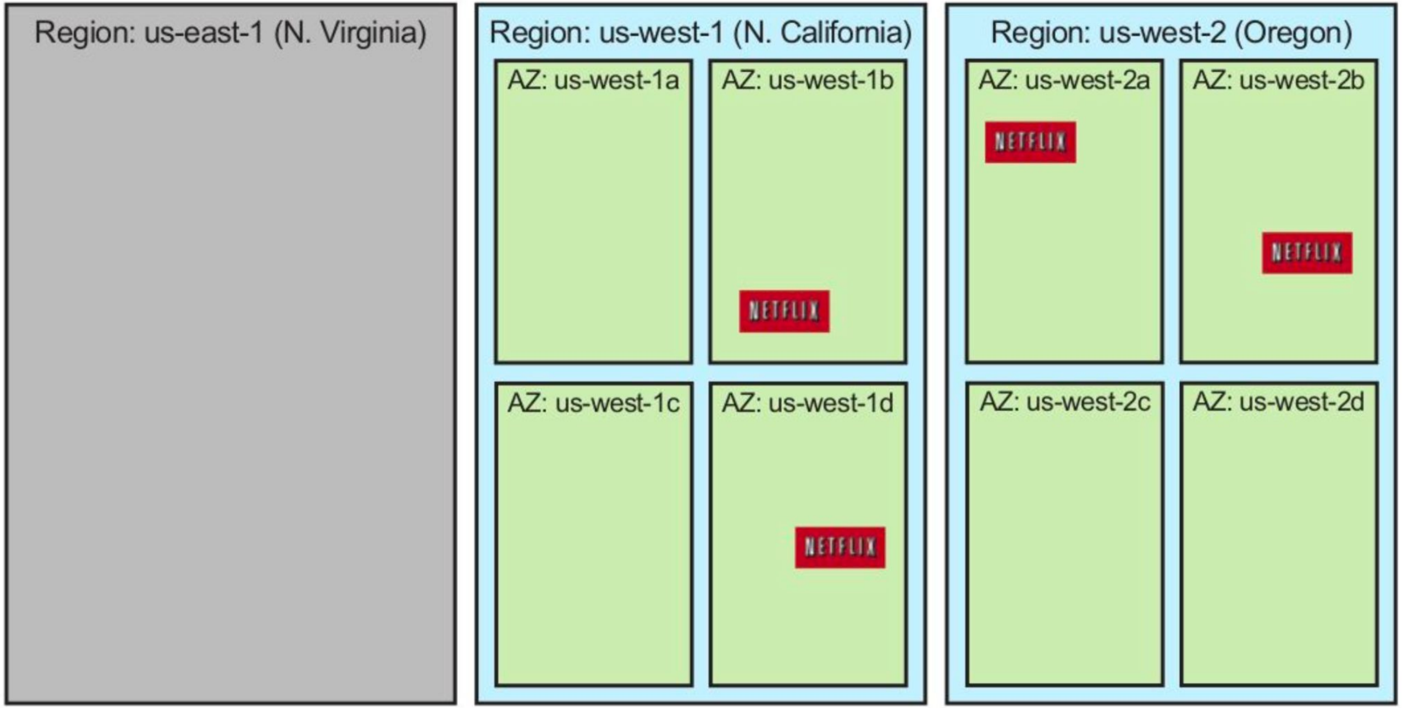
1. The Applications
2. The Containers
3. The Operating System
4. **The Hardware**

What is Container Orchestration? (1)



Ref.: Cloud Native Patterns, Cornelia Davis
https://www.manning.com/books/cloud-native-patterns

What is Container Orchestration? (2)



Ref.: Cloud Native Patterns, Cornelia Davis
https://www.manning.com/books/cloud-native-patterns

What is Container Orchestration? (3)

Deployment

"I want three of my
Node.js app Pods running"

Kubernetes

Kubernetes is the standard de facto for container orchestration

Can orchestrate Docker containers but it can use also other container engines

Usually it orchestrate a cluster of nodes, called worker nodes

Kubernetes has many objects, the simple and fundamental ones are:

- Pods
- Deployments
- Services

https://kubernetes.io/

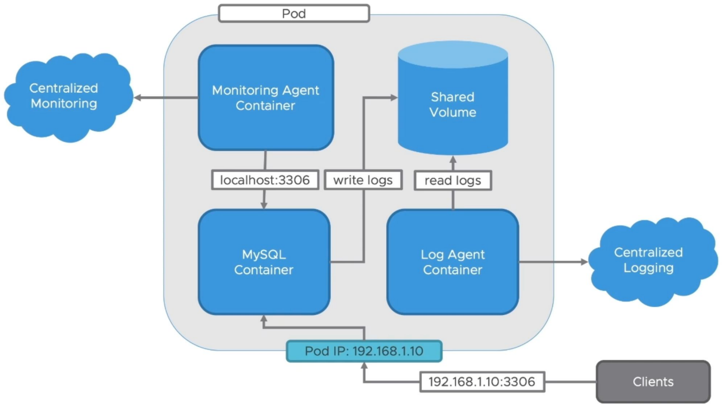
Kubernetes

Pod is the littlemost deployable unit on Kubernetes

Is a group of one or more tightly coupled containers

The Pod content is co-located, co-scheduled

Containers within the same Pod can communicate through localhost



Kubernetes

Kubernetes is the standard de facto for container orchestration

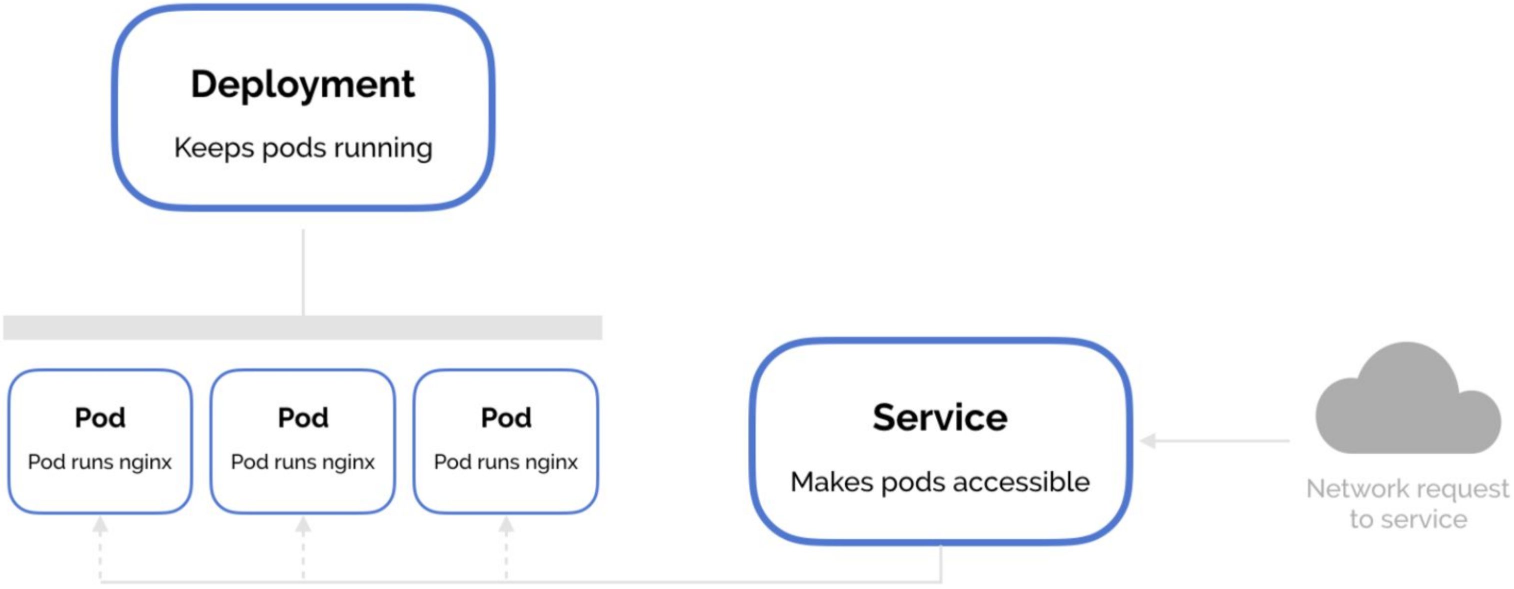
Can orchestrate Docker containers but it can use also other container engines

Usually it orchestrate a cluster of nodes, called worker nodes

Kubernetes has many objects, the simple and fundamental ones are:

- Pods
- Deployments
- Services

Kubernetes Pod, Deployment, and Service workflow




CMA CGM
CMA CGM
tex

Kubernetes for RAN Orchestration

Why Kubernetes for RAN?

- Standard for container orchestration in cloud-native environments.
- Efficient resource management for containerized workloads.
- Flexibility to deploy Monolithic, Disaggregated, and CUPS RAN architectures.

Energy Efficiency in RAN with Kubernetes:

- Kepler and Scaphandre integrate with Prometheus for energy observability.
- Real-time power consumption metrics enable optimization of workloads:
    - Orchestrating virtualized RAN components for edge and cloud
    - Energy-aware scheduling for dynamic and efficient resource allocation

Motivation for Energy Studies in RAN

Figure 15

Top reasons for measuring energy efficiency

How important are the following factors in measuring energy-efficiency indicators for your company?

More important

1. Track our company's progress over time (including targets)
2. Better understand and manage energy costs
3. To comply with reporting requirements/regulations
4. To include them in voluntary reporting and disclosures
5. Compare networks within our company
6. Compare with other operators/networks
7. Other (energy mix, finding optimum capex/opex relation)

Source: GSMA Intelligence Workshop Operator Survey 2024

Motivation for Energy Studies in RAN

- General Assembly of United Nations
  - "A better and more sustainable future for all"
- Key contribution
  - Information and Communication Technologies (ICT)
    - 5G & Beyond (B5G) networks
- B5G aims
  - Energy consumption ↓
  - Carbon footprint ↓

3% of all global energy consumed by ICT [1]

170% predicted jump in the ICT energy consumption by 2026 [1].

Radio Access Network (RAN) ⇒ 75% of energy consumption in B5G

[1] C. Freitag, M. Berners-Lee, K. Widdicks et al., "The real climate and transformative impact of ict: A critique of estimates, trends, and regulations," Patterns, vol. 2, no. 9, 2021.

RAN Energy Measurement Tools

How to measure Energy Consumption

Measuring energy consumption in Radio Access Networks (RAN) is crucial for understanding and improving efficiency

Energy can be measured in two main ways:

- **Hardware-based measurements**: Direct or indirect physical measurements
- **Software-based estimations**: Power consumption is approximated based on system data

Both methods have specific tools, advantages, and limitations

Hardware-Based Energy Measurements

Hardware tools **directly or indirectly** measure energy consumption.

- **Smart Plugs**: Affordable and less professional tools.
  - Example: Meross Smart Plug
  - Data can be accessed via:
    - MQTT Broker
    - Proprietary framework

- **Professional Power Meters**: High-precision measurement tools.
  - Example: GW Instek GPM-8213
  - Communication via **SCPI** (Standard Commands for Programmable Instruments)

Software-Based Energy Estimations

Software tools provide **energy consumption estimates** rather than direct measurements.

These estimates rely on system-level metrics (e.g., CPU usage, temperature, power states, CPU counters).

Examples of software tools:

- **S-TUI**: Lightweight terminal-based CPU monitoring.
- **Kepler**: Kubernetes-based energy estimation for workloads.
- **Scaphandre**: Power consumption export tool for cloud and edge systems.

Hardware vs Software: A Quick Comparison

Aspect	Hardware Measurement	Software Estimation
Accuracy	High (precise physical data)	Lower (based on estimations)
Cost	Variable (can be expensive)	Generally low-cost or free
Ease of Use	Requires setup and calibration	Easy to deploy on systems
Examples	Meross Plug, GW Instek GPM-8213	S-TUI, Kepler, Scaphandre

Prometheus - A Monitoring and Data Collection Tool

Prometheus is an open-source **monitoring and alerting** tool.

It collects, stores, and queries **time-series data**.

Designed for **cloud-native** environments and **microservices** architectures.

Key features:

- Pull-based data collection model.
- Multi-dimensional data model with labels.
- Integration with visualization tools like **Grafana**.

https://github.com/prometheus-operator/kube-prometheus

Prometheus Integration with Kepler and Scaphandre

Both **Kepler** and **Scaphandre** use Prometheus for:

- **Data Collection**: Metrics are exported in a format readable by Prometheus.
- **Storage and Querying**: Prometheus stores energy-related metrics for analysis.
- **Visualization**: Prometheus data can be visualized using tools like **Grafana**.

https://sustainable-computing.io/
https://github.com/hubblo-org/scaphandre

Prometheus Integration with Kepler and Scaphandre

Kepler:

- Exports **energy-related metrics** of Kubernetes workloads.
- Integrates with Prometheus to enable observability of power consumption.

Scaphandre:

- Exports **power consumption metrics** of systems (cloud/edge).
- Prometheus pulls these metrics for centralized analysis.

Grafana - Visualizing Energy Metrics

Grafana

Grafana is an open-source platform for monitoring and visualizing time-series data.

It integrates seamlessly with **Prometheus** to display energy-related metrics collected by tools like Kepler and Scaphandre.

Key Features:

- Customizable **dashboards** with real-time visualization.
- Supports multiple data sources (Prometheus, InfluxDB, etc.).
- Alerts and notifications based on thresholds.

Use Case:

- Energy metrics (e.g., power consumption, CPU usage) can be visualized and analyzed over time.
- Enables monitoring and decision-making for **RAN energy efficiency**.

https://grafana.com/

Stress Terminal UI (S-tui)

- Open-source software utility for stress-testing and monitoring system performance.
- Designed for terminal or CLI environments.
- **Purpose and Use Cases:**
  - Simulates and assesses system behavior under heavy load conditions.
  - Useful for testing and optimizing hardware and software configurations.
- **Features:**
  - User-friendly interface for real-time monitoring of system resource usage.
  - Displays key metrics:
    - CPU Utilization: Tracks usage across individual cores and averages.
    - Temperature: Monitors CPU temperature for various components.
    - Frequency: Reports CPU frequency (cycles per second).
    - Power: Monitors system power consumption (in Watts).
- **Benefits:**
  - Identifies potential bottlenecks, overheating issues, or system instability.
  - Helps in evaluating system response to high computational workloads.

https://github.com/amanusk/s-tui

Choosing the Right Tool

Hardware-based tools:

- **High precision of the whole energy consumption**
- **Consume more power, outside the under measurement system**
- **Needs physical access**

Software-based tools:

- **Useful for quick, low-cost estimations**
- **Has lower accuracy**
- **Adds overheads in the system**

Hybrid approaches: Combining both can offer a balance of accuracy and cost-effectiveness.

Scaphandre

- Key Features
  - **Power Monitoring Components:**
    - Tracks power consumption using Intel's RAPL sensors for hardware components like CPUs and GPUs.
    - Resource utilization metrics (CPU/memory) correlated with power data.
    - Process-level tracking with fine-grained intervals (jiffies).
  - **Power Metrics:**
    - Host Power (microwatts) ⇒ Aggregates RAPL and other sources for system-wide consumption.
    - Process Power (microwatts) ⇒ Tracks individual process consumption with labels (e.g., exe, pid).
    - Host Energy (microjoules) ⇒ Measures total energy usage.
    - CPU Frequency (MHz) ⇒ Monitors global CPU frequencies.
  - **Distinctive Capabilities:**
    - Measures per-process power consumption using timesharing and energy counters.
    - Real-time monitoring enabled by PowercapRAPL sensors and Linux kernel modules.
  - **Benefits:**
    - Delivers accurate, real-time insights into process-level power usage.
    - Ideal for scenarios involving multi-processor systems or VMs.

https://github.com/hubblo-org/scaphandre

Scaphandre

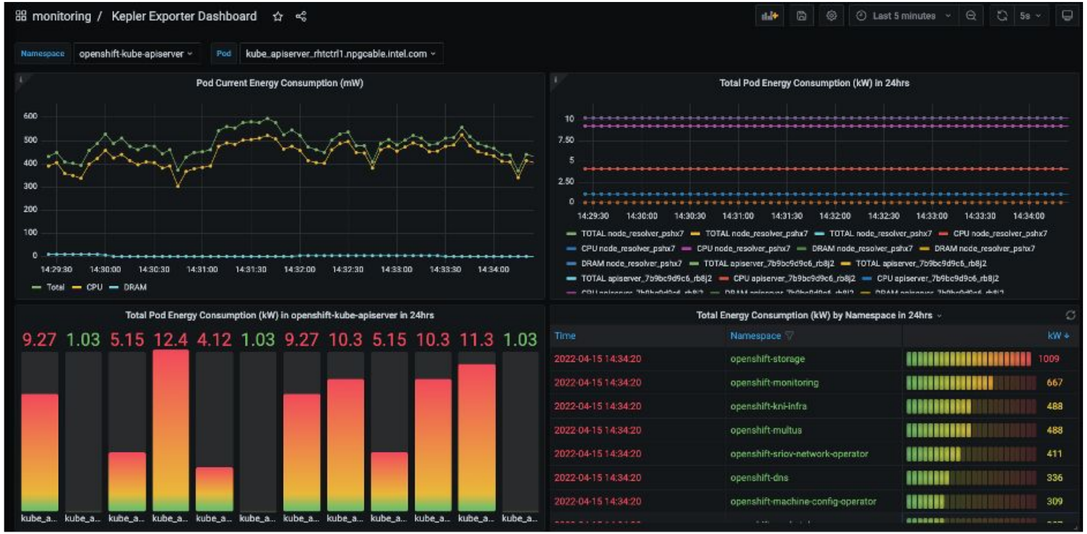
https://github.com/hubblo-org/scaphandre

Kubernetes Efficient Power Level Exporter (KEPLER)

- Developed by Red Hat, with contributions from IBM Research and Intel.
- Focused on power consumption in container cloud infrastructures.
- Key Features:
  - Leverages RAPL, ACPI, and NVIDIA GPU metrics for power measurements.
  - Based on eBPF for granular power attribution to processes, containers, and K8s pods.
  - **Functional Capabilities:**
    - Converts power data into energy estimates using custom ML-trained models.
    - Provides real-time metrics via Prometheus for analysis.
  - **Key Metrics:**
    - Container Core Joules Total ⇒ Power consumption of container CPU cores.
    - Container CPU Cycles Total ⇒ Total CPU cycles used by containers.
    - Container CPU Instructions Total ⇒ Total CPU instructions executed by containers.
    - Node Core Joules Total ⇒ Aggregated node power consumption.
  - **Use Case:**
    - Efficient power management for K8s clusters with insights into energy usage across workloads.
    - Supports sustainability goals in cloud infrastructures.

https://github.com/sustainable-computing-io/kepler

Kubernetes Efficient Power Level Exporter (KEPLER)



Demo Setup
MONOLITHIC, DIS-AGGREGATED, CUSP
OAI ARCHITECTURE

Our Reference Architecture (1)

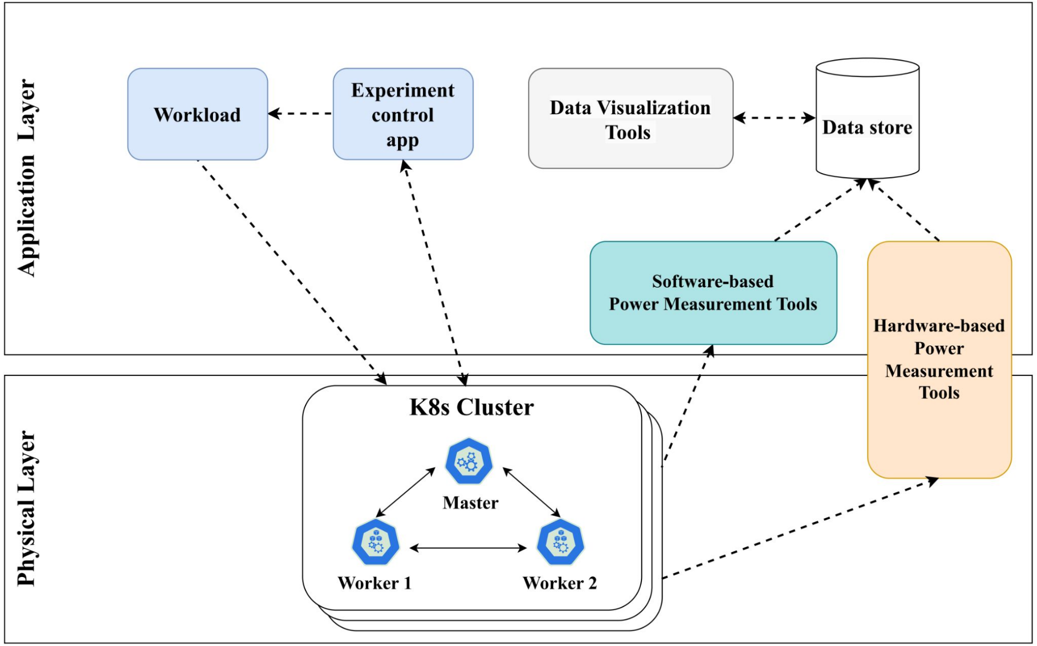
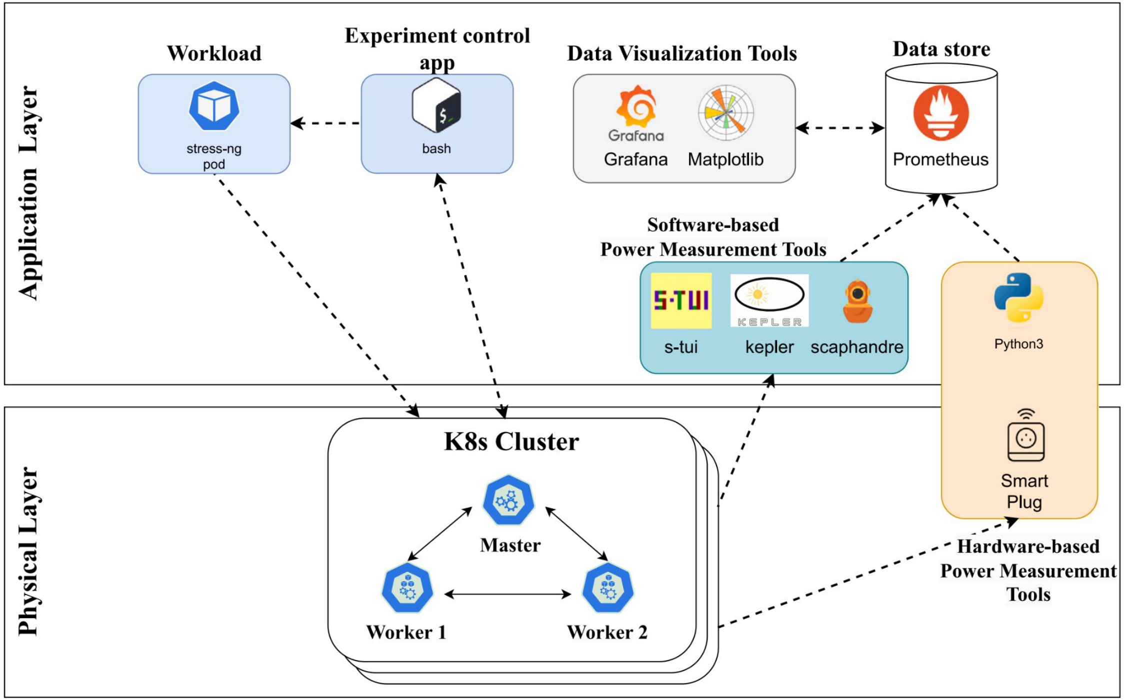


Fig. 1. High-level view of the Energy-efficient K8s-based infrastructure.

Centofanti, Carlo, et al. "Impact of power consumption in containerized clouds: A comprehensive analysis of open-source power measurement tools." Computer Networks 245 (2024): 110371.

Our Reference Architecture (2)

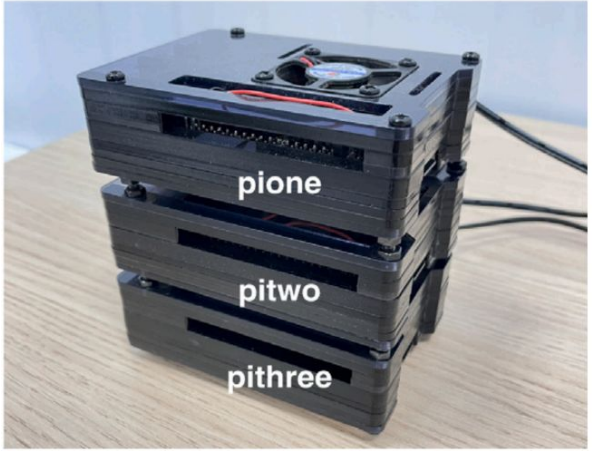


Centofanti, Carlo, et al. "Impact of power consumption in containerized clouds: A comprehensive analysis of open-source power measurement tools." Computer Networks 245 (2024): 110371.

Possble implementations

(a) *Virtual Wall (vWall)* testbed [36].

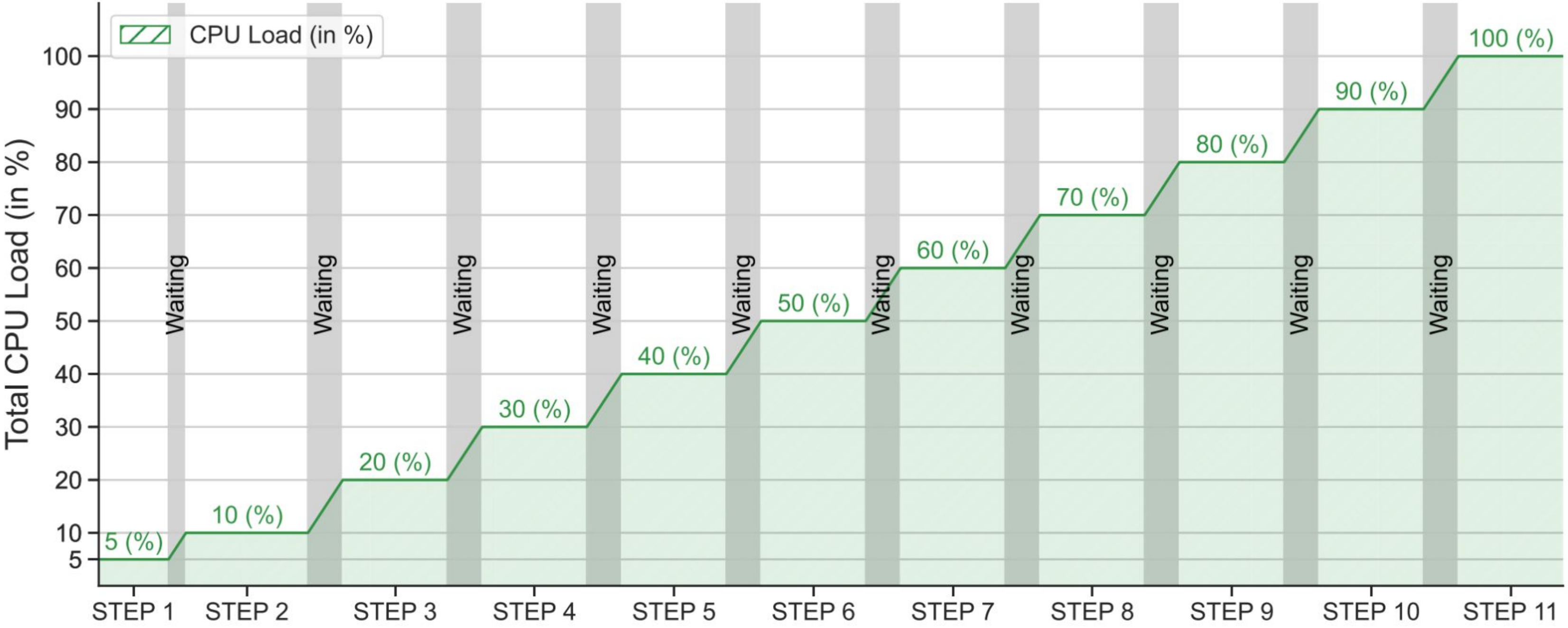
(b) *eCluster* testbed.



(c) *piCluster* testbed.

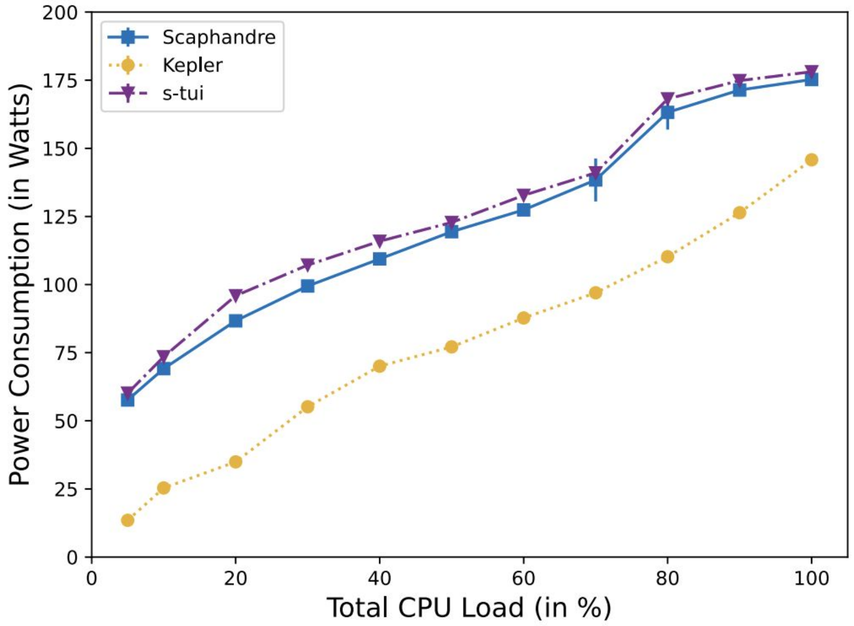
Centofanti, Carlo, et al. "Impact of power consumption in containerized clouds: A comprehensive analysis of open-source power measurement tools." Computer Networks 245 (2024): 110371.

Workload experiment

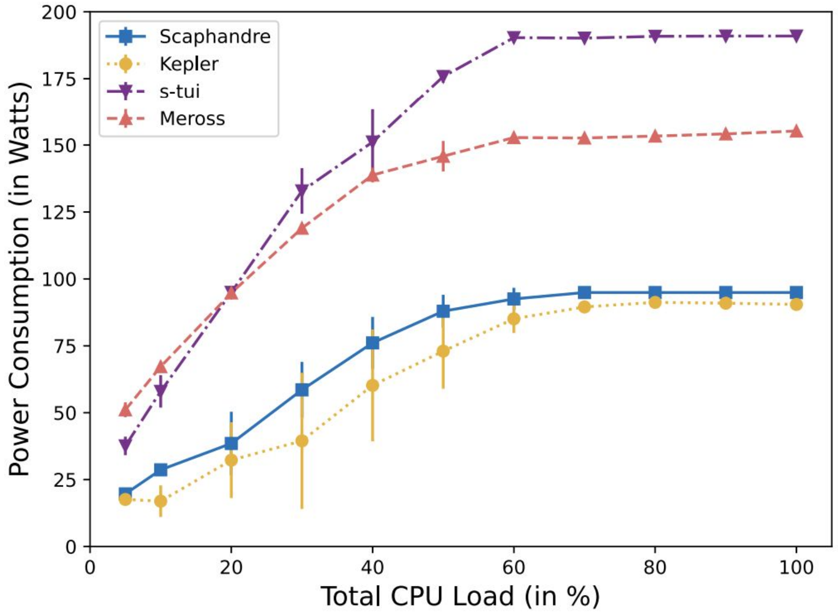


Centofanti, Carlo, et al. "Impact of power consumption in containerized clouds: A comprehensive analysis of open-source power measurement tools." Computer Networks 245 (2024): 110371.

Findings



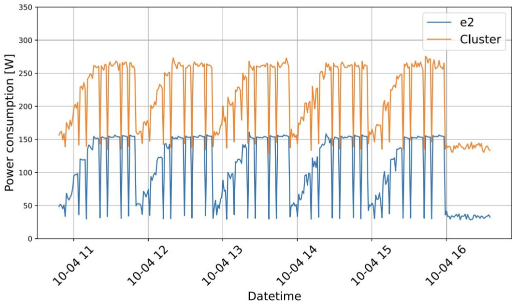
(a) *Virtual Wall (vWall)* testbed.



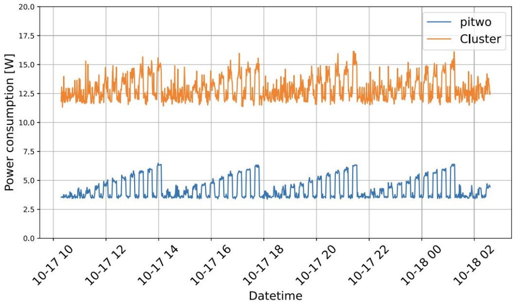
(b) *eCluster* testbed.

Centofanti, Carlo, et al. "Impact of power consumption in containerized clouds: A comprehensive analysis of open-source power measurement tools." Computer Networks 245 (2024): 110371.

Findings



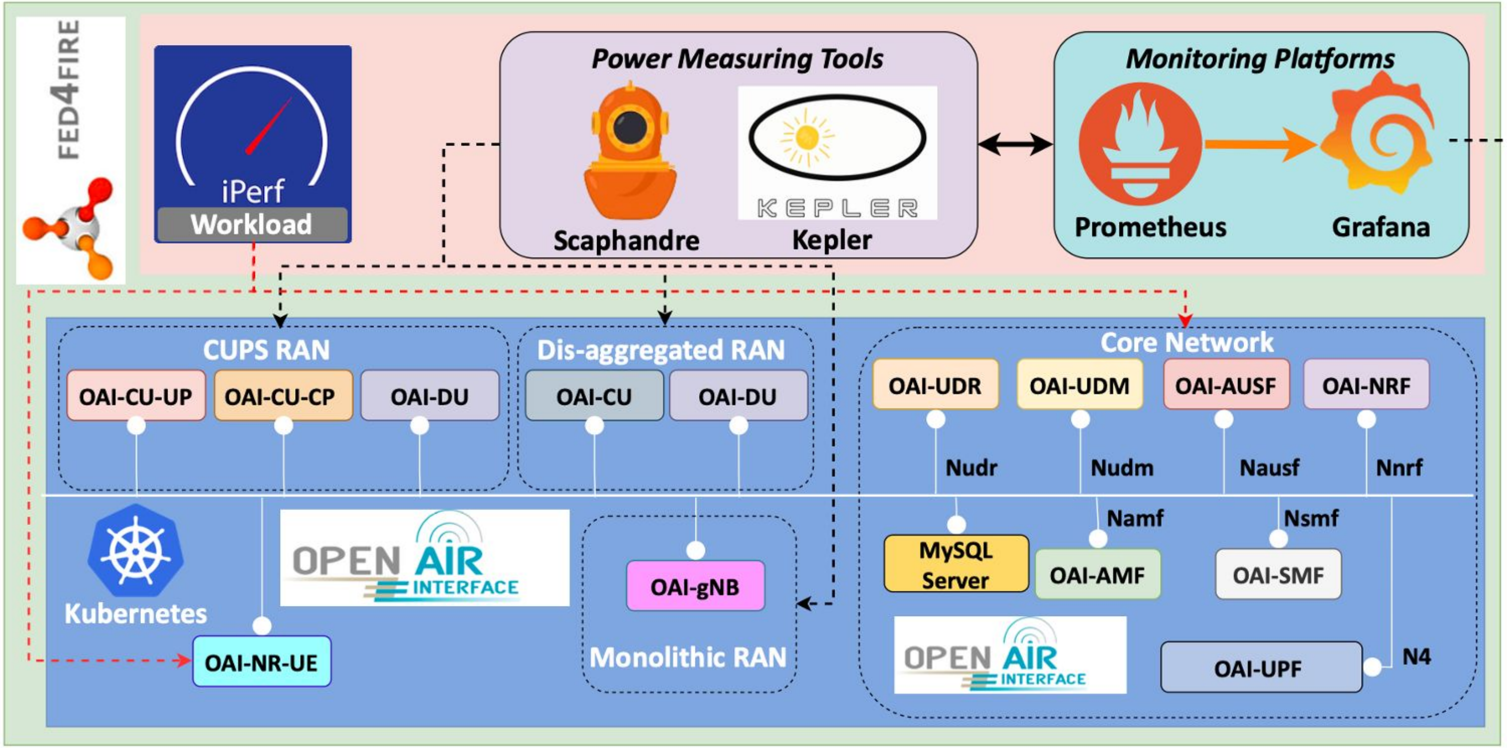
(a) *eCluster* testbed.



(b) *piCluster* testbed.

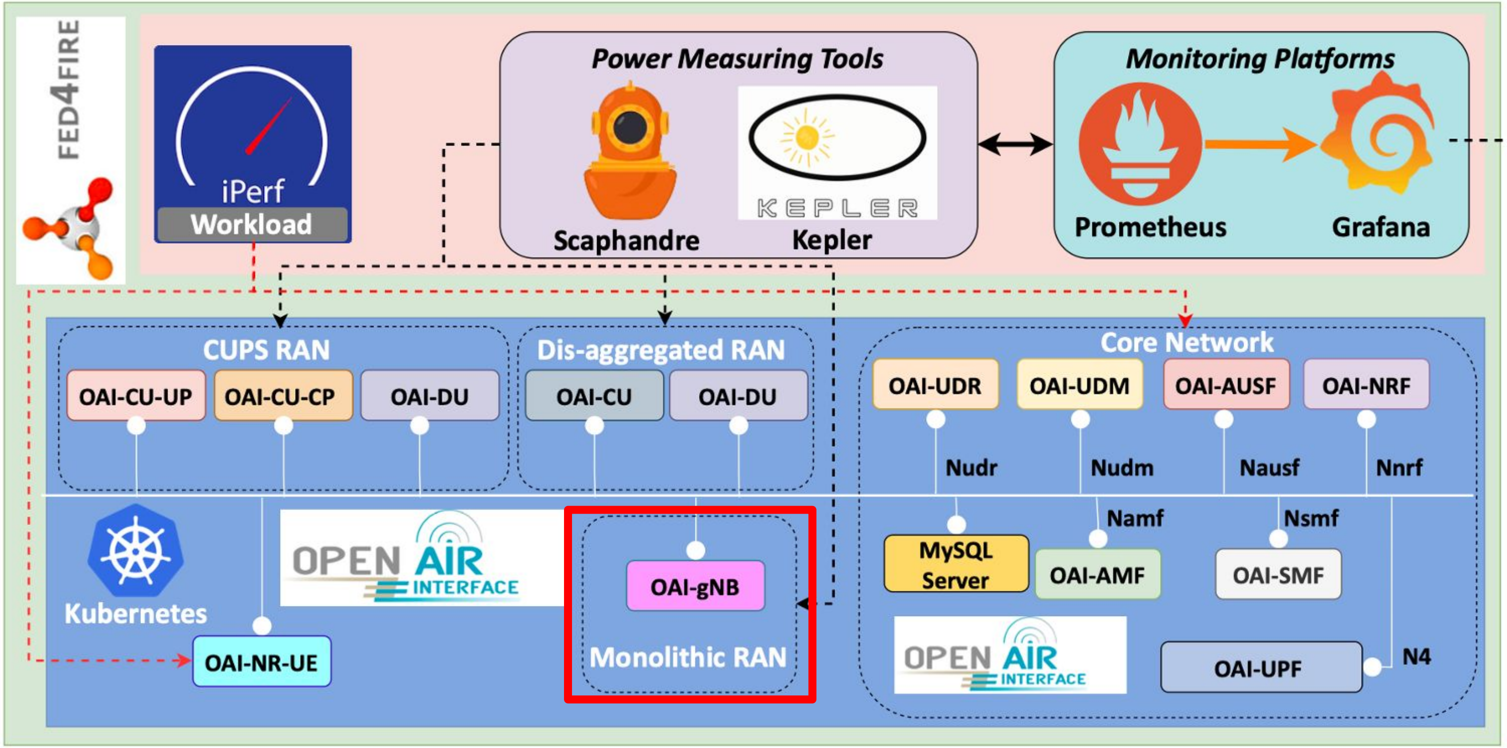
Centofanti, Carlo, et al. "Impact of power consumption in containerized clouds: A comprehensive analysis of open-source power measurement tools." Computer Networks 245 (2024): 110371.

Demo Reference Architecture (3)



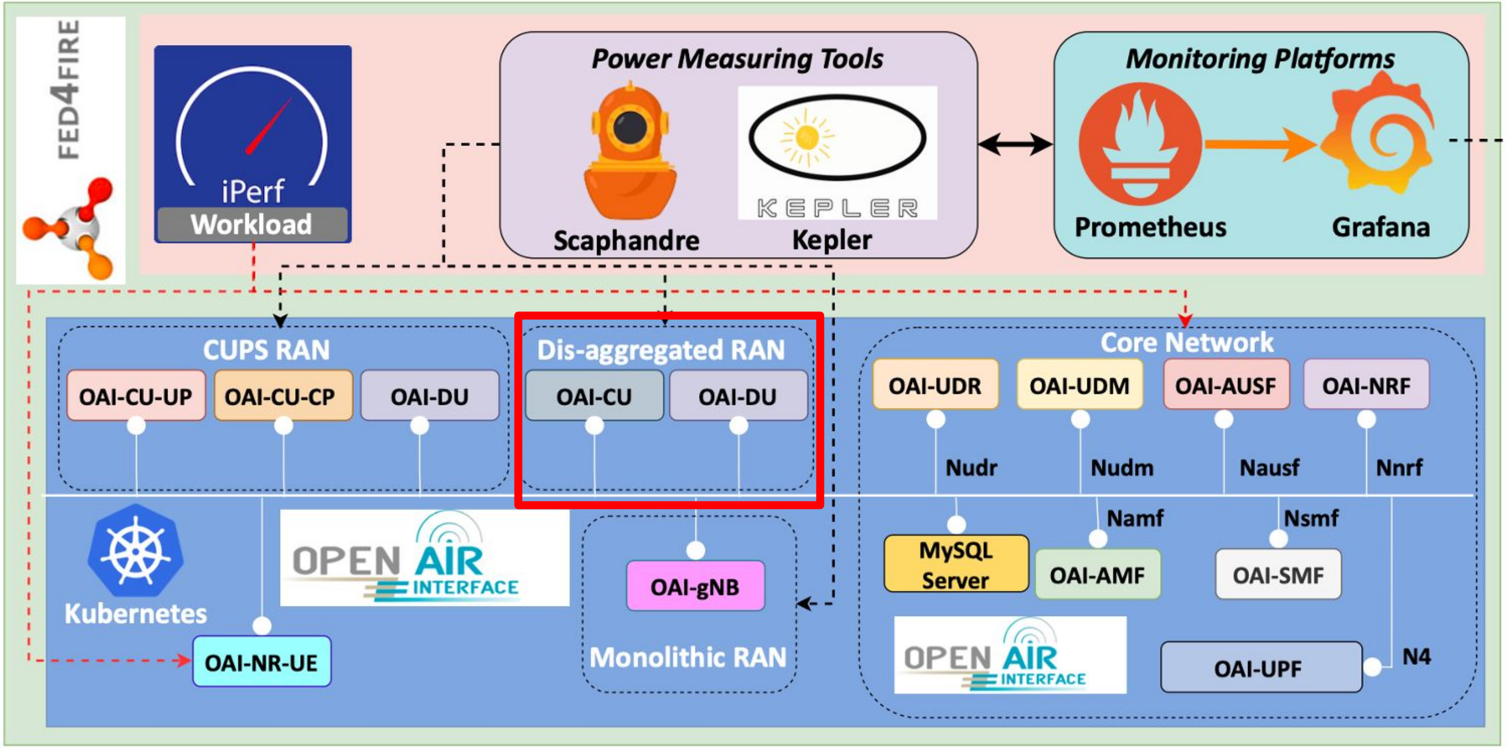
Gudepu, Venkateswarlu, et al. "Demonstrating the Energy Consumption of Radio Access Networks in Container Clouds." NOMS2024, the IEEE/IFIP Network Operations and Management Symposium. 2024.

Demo Reference Architecture - Monolithic

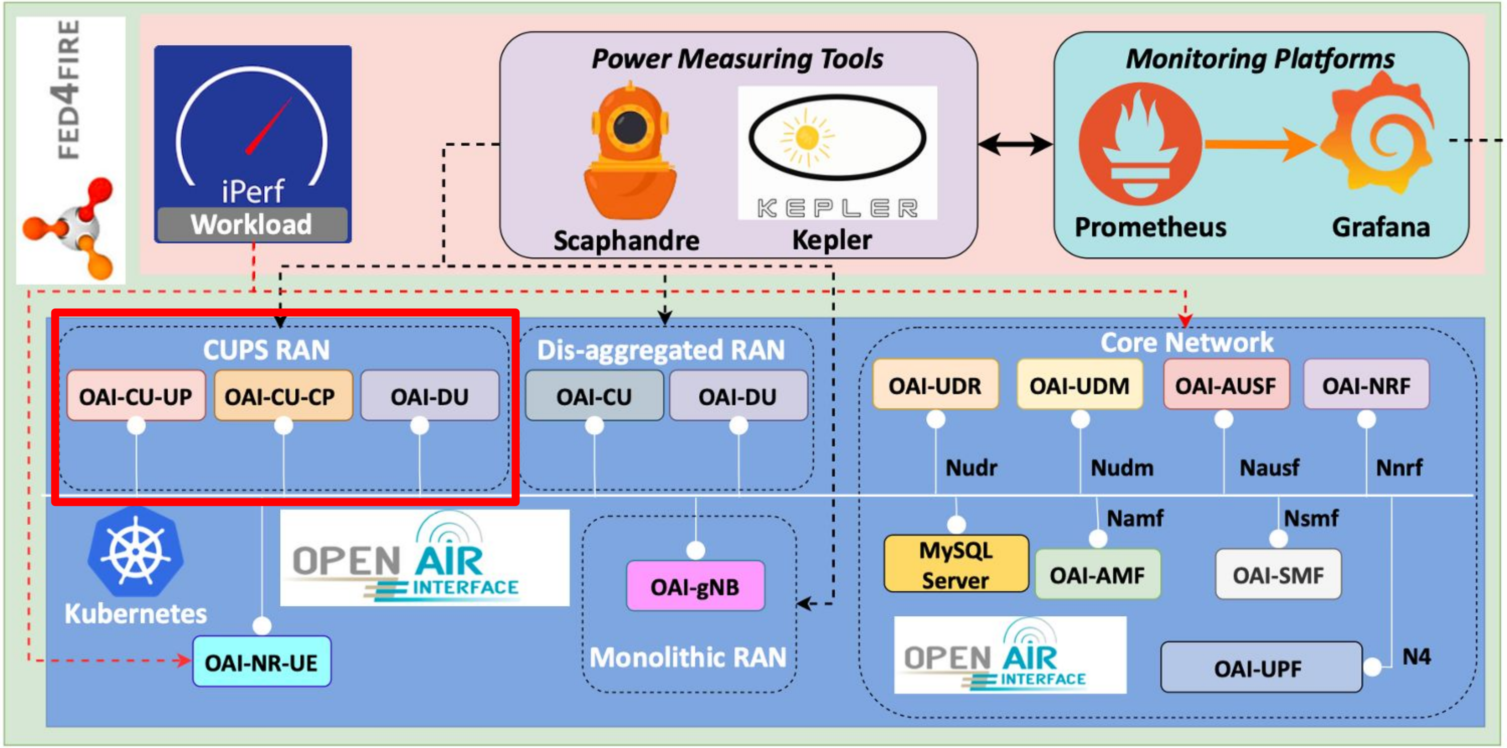


Gudepu, Venkateswarlu, et al. "Demonstrating the Energy Consumption of Radio Access Networks in Container Clouds." NOMS2024, the IEEE/IFIP Network Operations and Management Symposium. 2024.

Demo Reference Architecture - disaggregated



Demo Reference Architecture - CUPS



Gudepu, Venkateswarlu, et al. "Demonstrating the Energy Consumption of Radio Access Networks in Container Clouds." NOMS2024, the IEEE/IFIP Network Operations and Management Symposium. 2024.

Demo Time!
Hands-on Session

Where to find the code?

GitHub

https://github.com/Mellgood/2024-ants-tutorial-epic-ran

PRE-REQUISITE

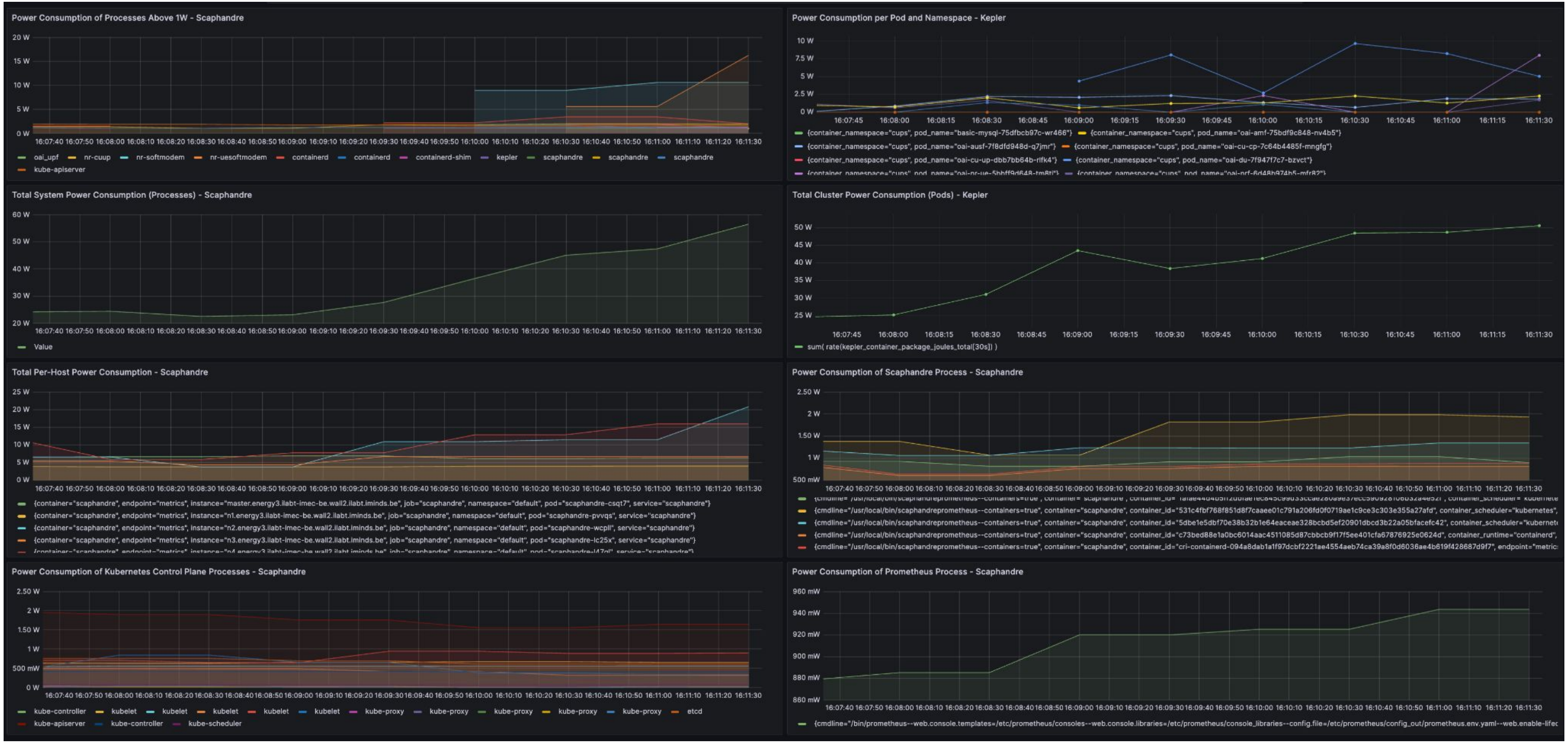
- Please ensure that you install the following for hands-on
- Docker
    - https://docs.docker.com/engine/install
- Kubernetes (k8s)
    - https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/create-cluster-kubeadm/
- Prometheus
    - https://github.com/prometheus-operator/kube-prometheus

Grafana Dashboard

We will monitor data from a Grafana Dashboard which for this tutorial will show:

- Power Consumption of Processes Above 1W - Scaphandre
- Power Consumption per Pod and Namespace - Kepler
- Total System Power Consumption (Processes) - Scaphandre
- Total Cluster Power Consumption (Pods) - Kepler
- Total Per-Host Power Consumption - Scaphandre
- Power Consumption of Scaphandre Process - Scaphandre
- Power Consumption of Kubernetes Control Plane Processes - Scaphandre
- Power Consumption of Prometheus Process - Scaphandre

Grafana Dashboard



Power Consumption of Processes Above 1W - Scaphandre

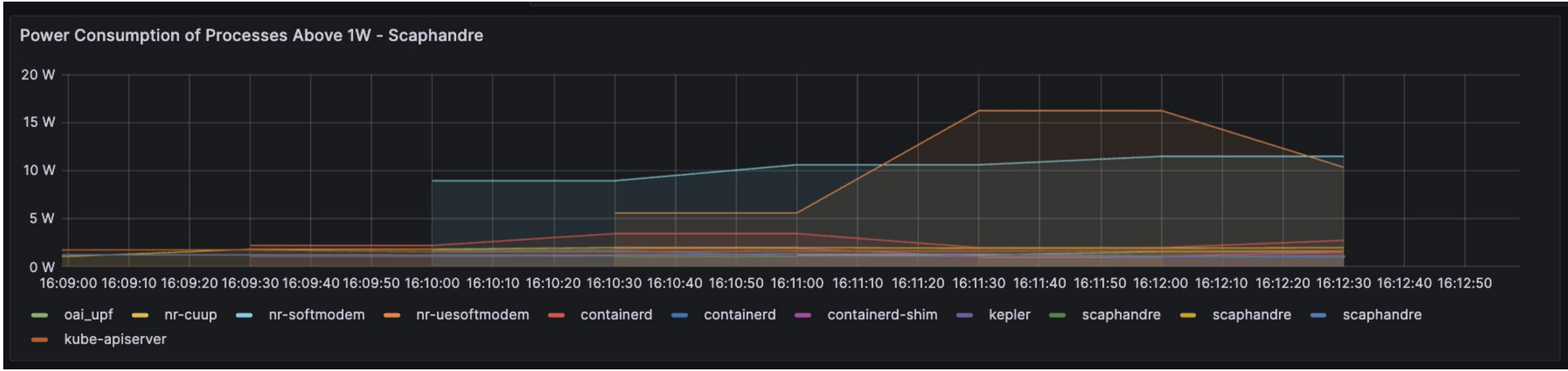
Query: scaph_process_power_consumption_microwatts / 1000000 > 1

Lines: Each line represents a monitored process

Helps focus on energy-intensive processes that can be optimized

Can detect anomalies or unexpected spikes in power usage

Power Consumption of Processes Above 1W - Scaphandre



Power Consumption per Pod and Namespace - Kepler

Query:

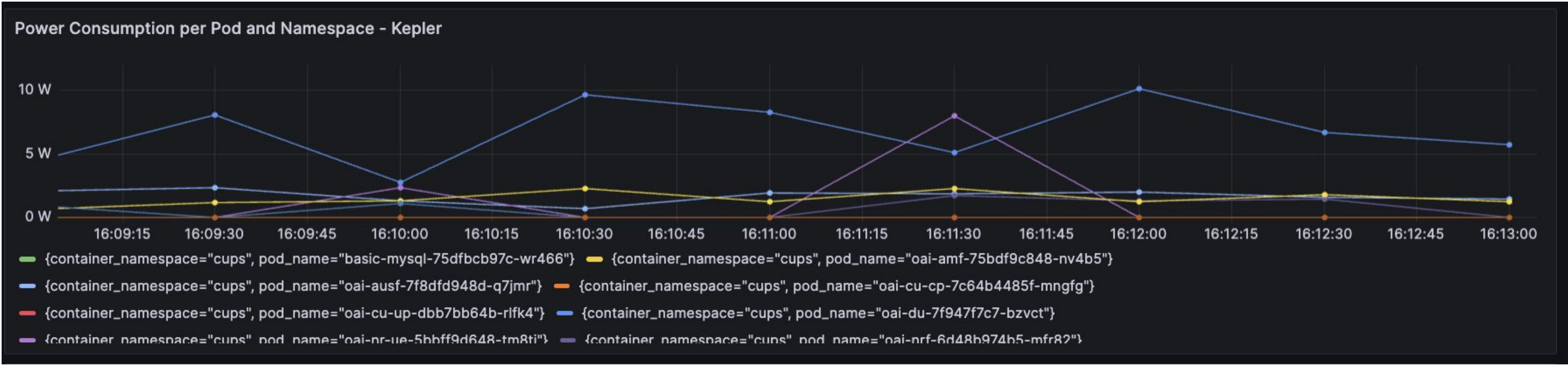
sum by (pod_name, container_namespace) (rate(kepler_container_package_joules_total{container_namespace=~"mono|dis|cups"}[30s]))

Lines: Each line represents a pod within one of the selected namespaces (mono, dis, cups)

Enables energy monitoring per namespace for targeted analysis

Can detect unbalanced workloads or energy inefficiencies across pods

Power Consumption per Pod and Namespace - Kepler



Total System Power Consumption (Processes) - Scaphandre

Query:

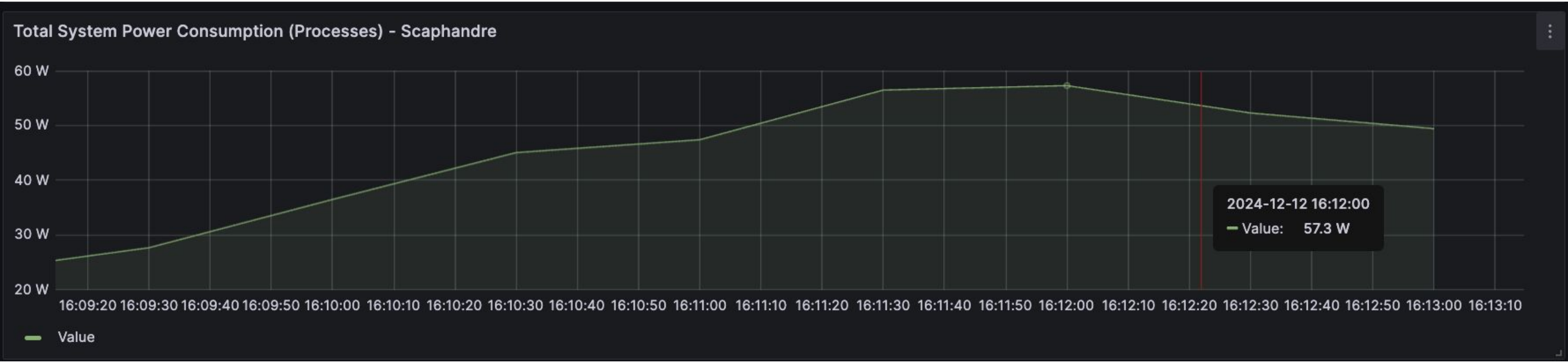
```
sum(scaph_process_power_consumption_microwatts / 1e6)
```

Lines: Represents the **total power consumption** of all monitored **processes** on the system

A global view of the system's energy usage

Helps determine if the system's overall power usage is within expected limits

Total System Power Consumption (Processes) - Scaphandre



Total Cluster Power Consumption (Pods) - Kepler

Query:

sum(rate(kepler_container_package_joules_total[30s]))

Lines: Represents the **total power consumption** of all Kubernetes **pods** in the cluster

The total energy usage of all pods over time

Useful for assessing the energy footprint of Kubernetes workloads

Total Cluster Power Consumption (Pods) - Kepler

Total Per-Host Power Consumption - Scaphandre

Query:

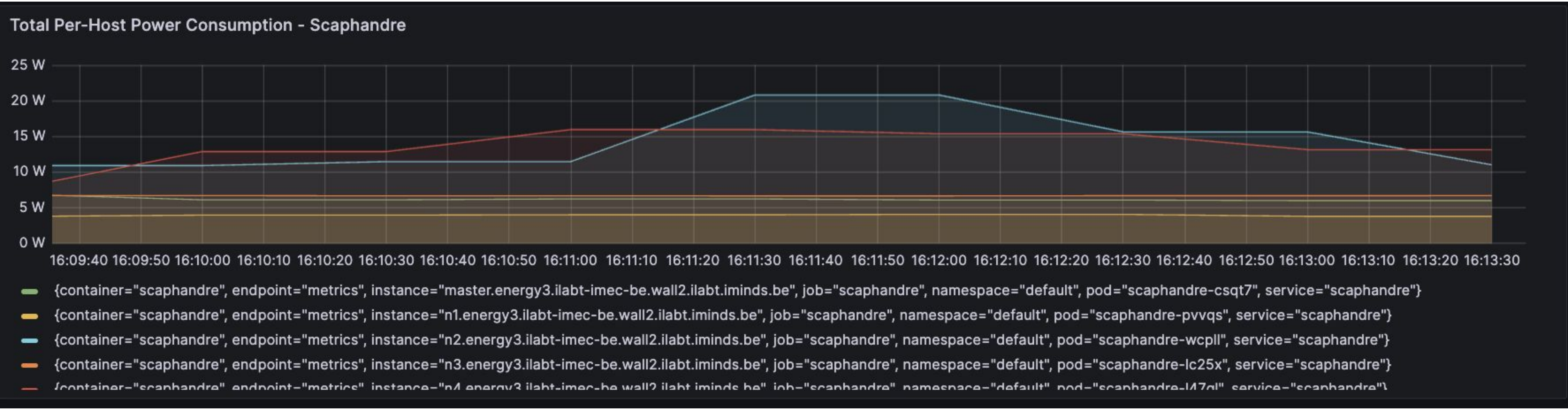
scaph_host_power_microwatts / 1000000

Lines: Represents the total power consumption of the host machine, as measured by Scaphandre

The overall energy consumption of the physical hosts over time

Monitor per-host energy consumption to establish baseline usage patterns

Total Per-Host Power Consumption - Scaphandre



Power Consumption of Scaphandre Process - Scaphandre

Query:

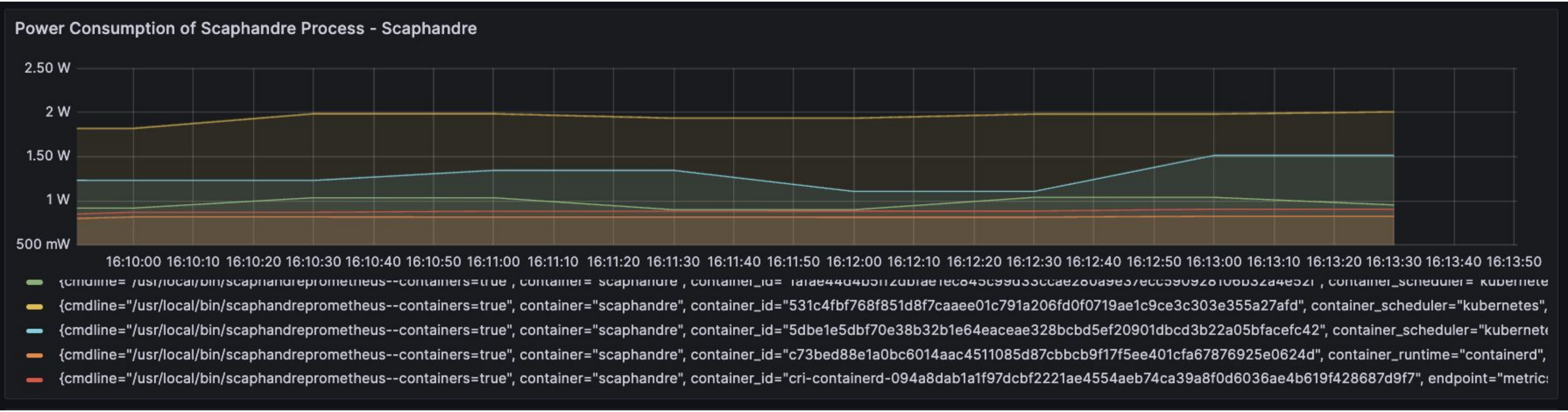
scaph_process_power_consumption_microwatts{exe="scaphandre"} / 1000000

Lines: Represents the power usage of the Scaphandre process

Even monitoring tools have an energy cost; this graph quantifies it

Scaphandre's power consumption should remain low and stable

Power Consumption of Scaphandre Process - Scaphandre



Power Consumption of Kubernetes Control Plane Processes - Scaphandre

Query:

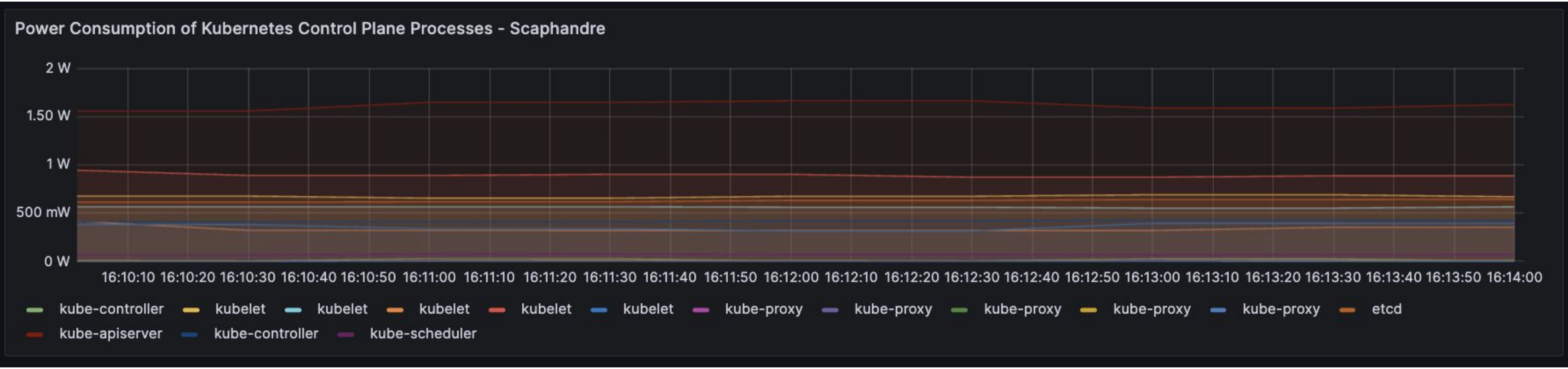
scaph_process_power_consumption_microwatts{exe=~"etcd|kube-apiserver|kube-controller|kube-proxy|kube-scheduler|kubelet"} / 1000000

Lines: Each line represents the power consumption of one specific Kubernetes process

Establishes baseline consumption for Kubernetes operations

Use these insights to optimize Kubernetes configurations for energy efficiency

Power Consumption of Kubernetes Control Plane Processes - Scaphandre



Power Consumption of Prometheus Process - Scaphandre

Query:

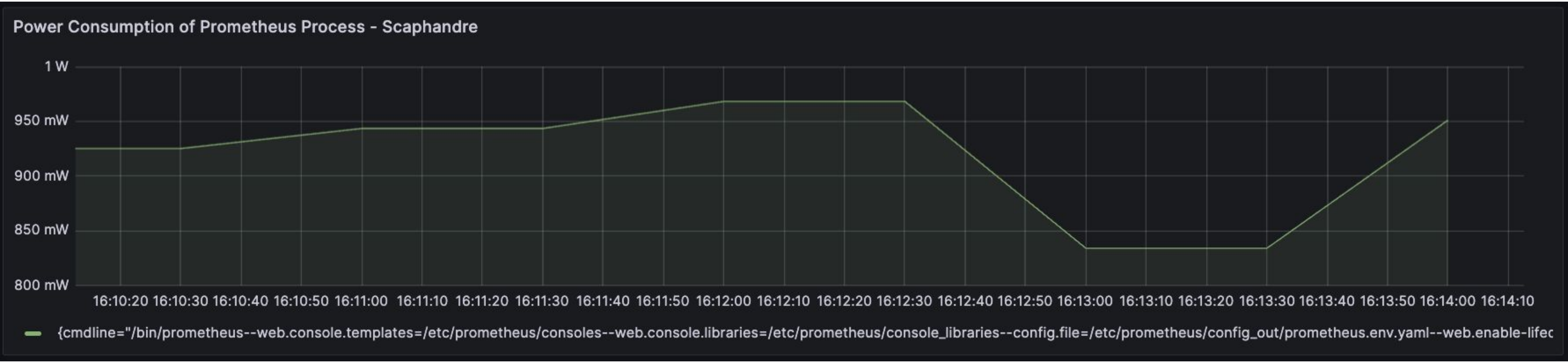
scaph_process_power_consumption_microwatts{exe="prometheus"} / 1000000

Lines: Represents the power usage of the Prometheus process over time

Prometheus itself consumes resources; this chart quantifies its energy footprint

High energy usage might indicate excessive scraping intervals, inefficient queries, or misconfigurations

Power Consumption of Prometheus Process - Scaphandre

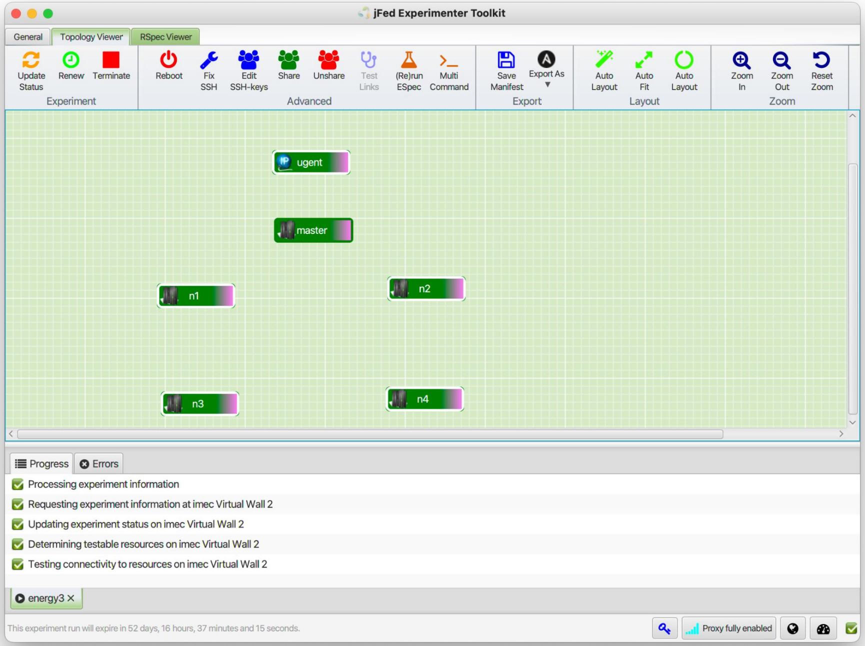


The Virtual Wall facilities (vWall)

Accessible through jFed

Research testbed born from the Fed4Fire European project

Remotely accessible configurable virtual lab



FED4FIRE
FEDERATION FOR FIRE PLUS

IEEE ANTS

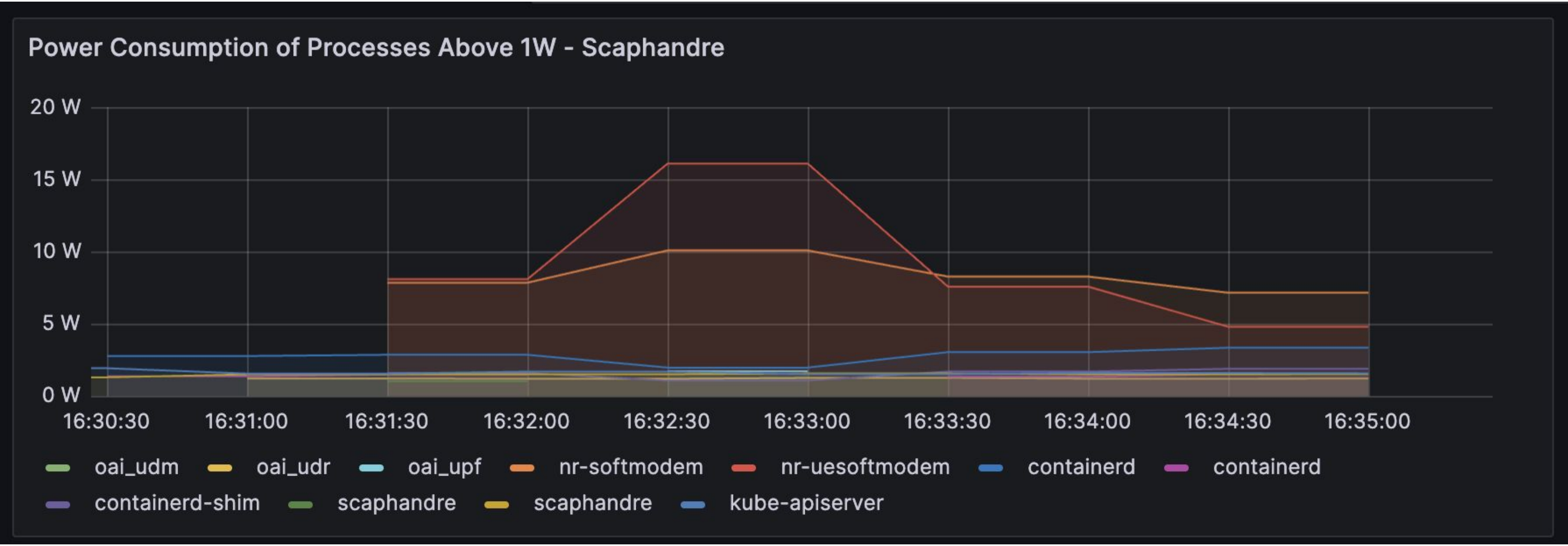
UNIVERSITÀ DEGLI STUDI DELL'AQUILA
IIT DHARWAD
GHENT UNIVERSITY

Let’s deploy Monolithic RAN

carlo — dany808@master: ~/OAI-5g_Deployment_HC-main/monolithic —...

```
dany808@master:~/OAI-5g_Deployment_HC-main$ cd monolithic/
dany808@master:~/OAI-5g_Deployment_HC-main/monolithic$ ls
charts          oai-scripts     start.sh  uninstall_oai.sh
deploy_oai.sh   oai-scripts_bk  stop.sh
dany808@master:~/OAI-5g_Deployment_HC-main/monolithic$ ./deploy_oai.sh mono
```

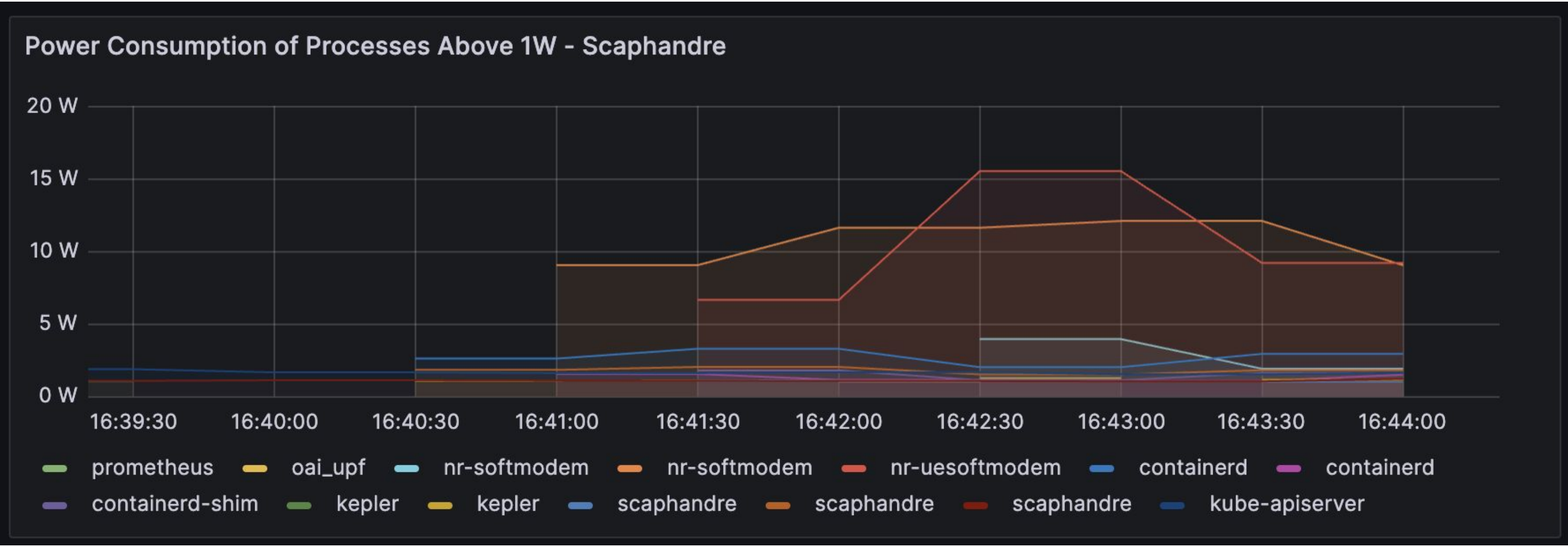
Grafana perspective - Monolithic



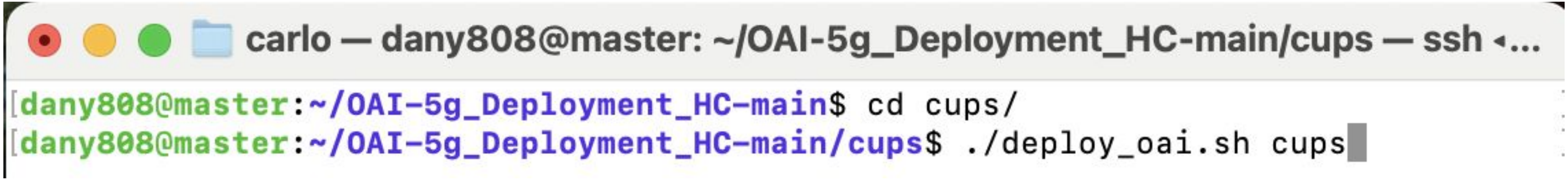
Let’s deploy Disaggregated RAN

```
carlo — dany808@master: ~/OAI-5g_Deployment_HC-main/disaggregat...
dany808@master:~/OAI-5g_Deployment_HC-main$ cd disaggregated/
dany808@master:~/OAI-5g_Deployment_HC-main/disaggregated$ ./deploy_oai.sh dis
```

Grafana perspective - Disaggregated



Let’s deploy CUPS RAN



Demo Setup

Environment setup for energy measurement in RAN

description of the testbed

Q&A Session

Interactive discussion with participants

Wrap-up

Conclusions (1)

Summary of Key Findings:

- Energy consumption is a critical challenge in containerized RANs.
- Monolithic, Disaggregated, and CUPS architectures offer trade-offs in flexibility, scalability, and energy efficiency.
- Tools like Scaphandre, Kepler, and Prometheus enable real-time monitoring and actionable insights.

Hands-On Insights:

- Demonstrated practical energy measurements across various RAN architectures.
- Grafana dashboards highlighted areas for energy optimization in Kubernetes environments.

Conclusions (2)

Future Directions:

- Advancing energy-efficient orchestration strategies.
- Integrating AI/ML for dynamic energy optimization in RANs.
- Addressing scalability challenges for large-scale deployments.

Call to Action

- Adopt energy-efficient tools and methodologies to enhance sustainability.
- Leverage containerized architectures for future-proof RAN deployments.

Thank You!

Carlo Centofanti
Assistant Professor
University of L'Aquila (Italy)

carlo.centofanti1@univaq.it