Vodafone OpenRAN O-Cloud Performance Benchmark Spring 2025

Content

1. Executive Summary

O-Cloud (CaaS – Container as a Service) provides an essential disaggregation role between the hardware and the application software enabling the necessary cloud computing capabilities to execute RAN (Radio Access Network) functions. Performance and maturity of this O-Cloud layer is important for the successful delivery of a high performing RAN.

Vodafone has delivered the industry's first O-Cloud (CaaS) performance benchmark, focusing on Container as a Service (CaaS) solutions within OpenRAN. This benchmarking exercise evaluated the performance of three Tier-1 CaaS vendors using the VoerEir´s "Touchstone" testing and benchmarking tool in Vodafone's Central Lab. Vodafone has conducted the performance benchmark with the purpose to gather data to evaluate industry state of art and compare vendors' performance to achieve best OpenRAN performance and operational stability for live deployment.

The benchmark covered 500+ test cases in various aspects, including Kubernetes Functional testing, Kubernetes Security Conformance testing, Compute Performance, Network Performance, Storage Performance, Control Plane Performance, CaaS Platform Robustness, High Availability and Stability testing. All three vendors were given 2 identical Dell XR8620t servers to build their CaaS cluster, and the testing was run in parallel to achieve fair comparison (except network performance testing to void network congestion).

Below figure illustrated all three vendors benchmark results, which highlighted strengths and weaknesses of each vendor. The color code present vendors' scoring of each testing category comparing with target value setup by Touchstone tool.

90-100% 70-89% 50-69% 0-49%

Testing Category	Vendor A	Vendor B	Vendor C
Compute Performance			
Network Performance - Throughput			
Network Performance - Jitter & Latency			
Storage Performance			
Control Plane Performance			
Robustness Testing			
High Availability			
Kubernetes Functionality			
Security Configuration Testing			

Table 1 – O-Cloud Vendors' Anonymized Benchmark Results

Our key findings from this first CaaS performance benchmark campaign include:

- Excellent upstream compatibility with Kubernetes functionality and security across all vendors.
- Excellent Kubernetes Control Plane performance, Robustness and High Availability demonstrated all vendors' CaaS provide solid platform resilience.

- Effective Compute performance and Network Performance for real-time OpenRAN DU/CU (Distributed Unit/Centralized Unit) data processing.

- Excellent Storage performance across 2 vendors, and only 1 vendor underscored due to lack of configuration of periodic housekeeping.

- Software only testing tools like the one used in this benchmark works efficiently well for this type of testing removing the need of having hardware testing tool for O-Cloud/CaaS testing purposes

In conclusion, Vodafone's pioneering efforts in CaaS performance benchmarking provide a foundation for ongoing evaluation and continuous improvement in OpenRAN O-Cloud implementations. This Vodafone´s initiative sets a precedent for rigorous benchmark standards and continuous optimization to enhance CaaS platform performance for OpenRAN.

2. O-Cloud Architecture and Container as a Service (CaaS) in OpenRAN

In the Open RAN (O-RAN) architecture (see figure 1), it consists of 3-layer components: Service Management and Orchestration (SMO), Open RAN Application layer and O-Cloud (Container as a Service-CaaS) Layer.

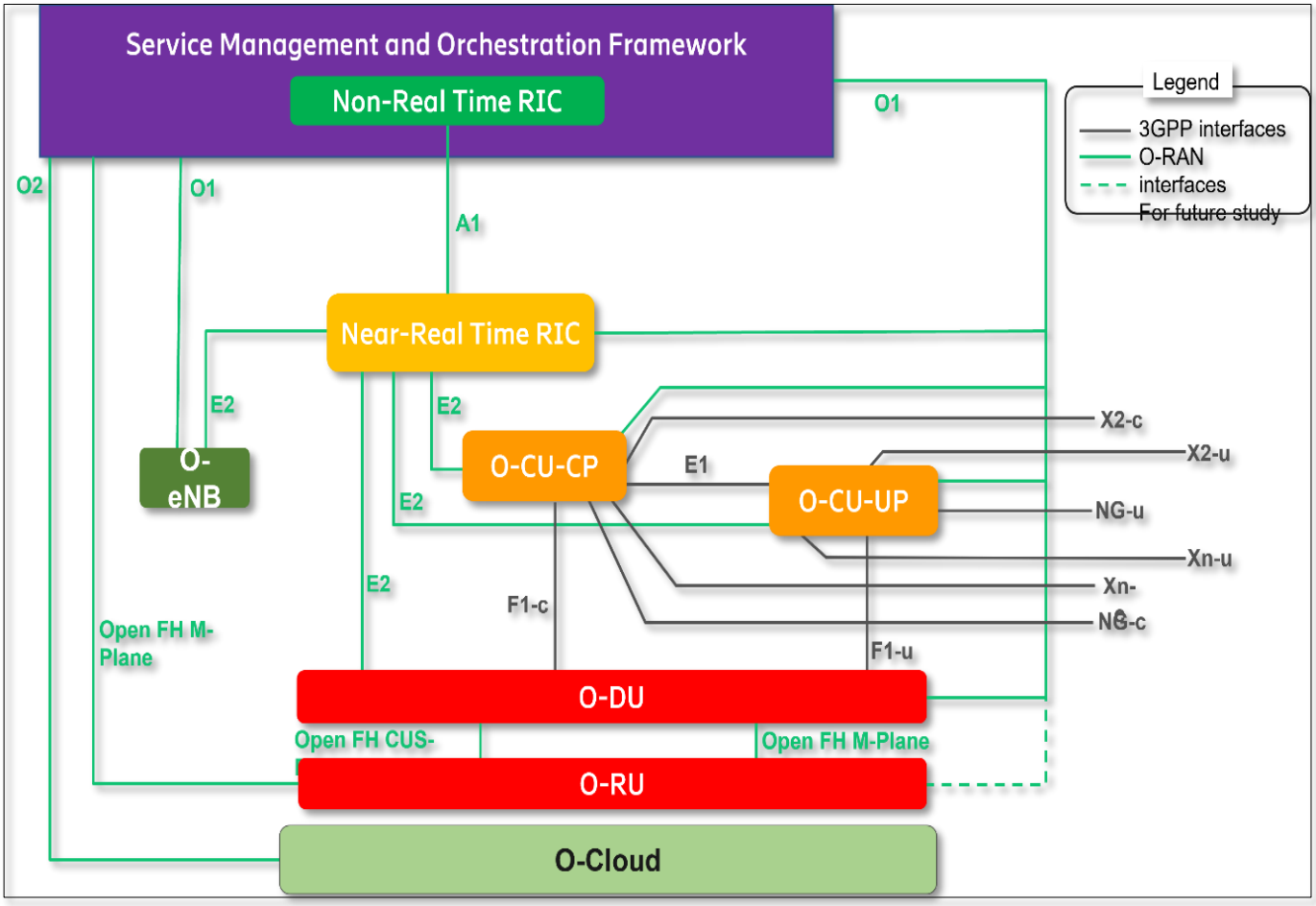


Figure 1- Open RAN Architecture in ORAN Alliance Specification

SMO consists of Network Function Orchestration (NFO) and Non-Real Time RAN Intelligence Controller (RIC) components. It is an intelligent automation platform which applies automation at scale to simplify the complexity of the networks, as well as improve network performance, enhance customer experience and minimize OpenRAN operational cost.

Open RAN application layer consists of O-RAN Radio Unit (O-DU), Distributed Unit (O-DU), and O-RAN Central Unit (O-CU). For O-CU, the functionality is divided into Control Plane (O-CU-CP) and User Plane (O-CU-UP). As Vodafone network architecture is Distributed RAN (DRAN), thus the O-DU and O-CU are deployed on the same server on remote edge RAN site. Meanwhile, Near-Real Time RIC can also be deployed at RAN site to optimize the network performance in real-time manner to achieve best network capacity & coverage.

O-Cloud (CaaS) layer serves as the underlying cloud computing platform, comprising a network of physical infrastructure nodes that adhere to O-RAN specifications. It is an essential component as it enables RAN software and hardware disaggregation and empowered RAN with open interface and multi-vendor solutions. O-Cloud (CaaS) provides Real Time Operating System (RT-OS) to support O-DU real time packet processing, it

also provides Time & Synchronization solution which is the foundation of DRAN deployment. Additionally, it integrates the required management and orchestration capabilities to ensure seamless operation automation and scalability. This open, interoperable infrastructure enhances network agility and cost efficiency.

Why O-Cloud (CaaS) is a key for OpenRAN?

O-Cloud (CaaS) is a critical component for managing cloud-native RAN applications. Kubernetes - the industry- standard container orchestration platform, is used to automate deployment, scaling, and operations of containerized O-RAN workloads.

CaaS solution is an essential enabler for OpenRAN because it disaggregates OpenRAN software from underlying hardware to allow containerized RAN applications to be implemented on General Purpose Processor (GPP) based Commercial Off the Shelf (COTS) servers. CaaS solution for OpenRAN is designed with small footprint i.e. Only 2 CPU Cores required to implement full Kubernetes functionality to support RAN software at edge OpenRAN site. CaaS solution also provide the important Real-Time Operating System (RT-OS) with a maximum latency less than 20µs, which is the key to support time critical O-DU packet processing. Meanwhile, CaaS platform implements the fundamental Time & Synchronization solution for edge RAN sites e.g. GNSS (Global Navigation Satellite System), Precision Time Protocol (PTP) synchronization or Synch E.

In this whitepaper, we'll mainly focus on the performance benchmark of CaaS solutions from industry Tier-1 CaaS suppliers to ensure CaaS platform provide the best performance to meet our network KPI (Key Performance Indicator).

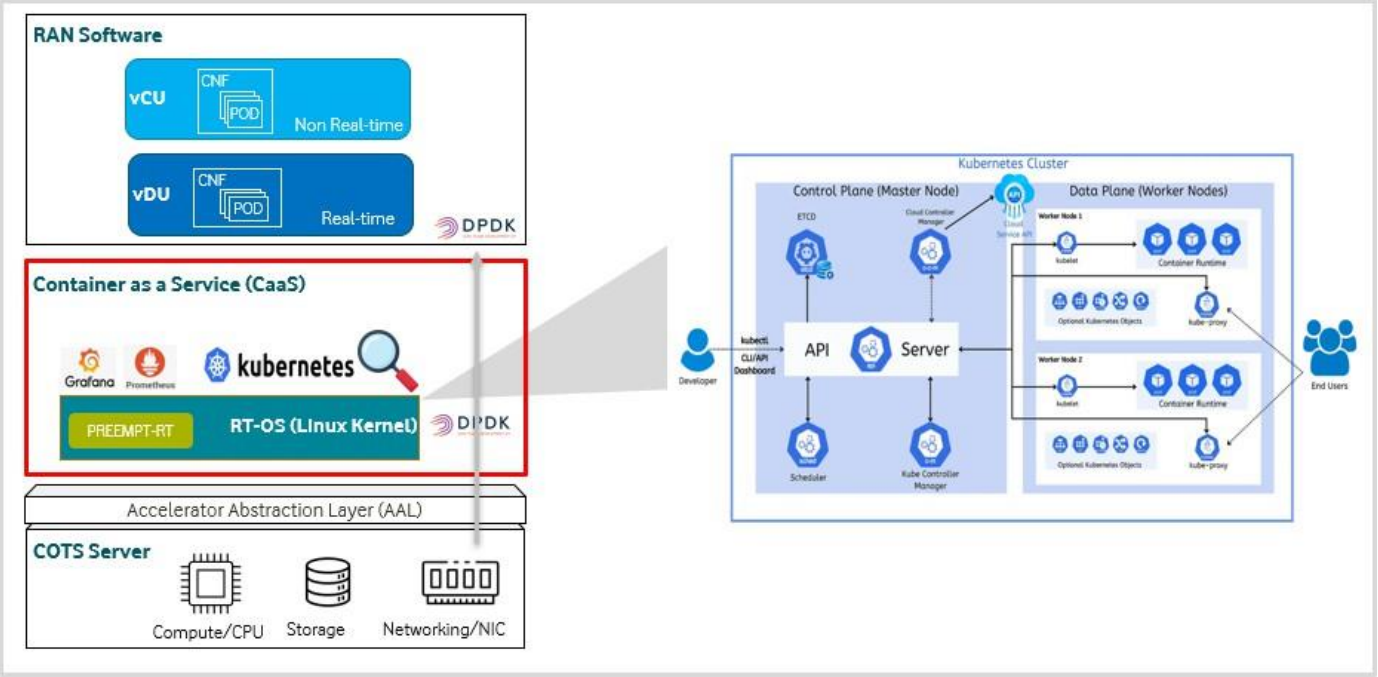


Figure 2 - O-Cloud Implementation for Vodafone DRAN Architecture

3. Test Methodology and Testing Tool

The purpose of this benchmarking exercise was to assess the CaaS platform across three vendors that will be kept anonymous in this document, focusing on Kubernetes functionality, CPU & Network & Storage performance, CaaS platform stability, CaaS fault- tolerance, security conformance. These performance tests were conducted in parallel to achieve fair comparison with the exception of network performance tests which were carried out sequentially to avoid the network congestion and any potential the impact on vendor results.

We introduced 3rd party VoerEir´s "Touchstone" a cloud system benchmarking and testing tool to carry out the testing and provide independent validation report. As shown in below figure, "Touchstone" test management system was installed on dedicated bare metal COTS server. "Test Agent" was deployed on CaaS cluster as application. In this benchmark, we invited three CaaS vendors to participate in the testing; thus there were three CaaS cluster shown in below figure. "Touchstone" tool generated generic UDP (User Datagram Protocol), TCP (Transmission Control Protocol), SCTP (Stream Control Transmission Protocol) traffic in Pod level to measure the CaaS performance.

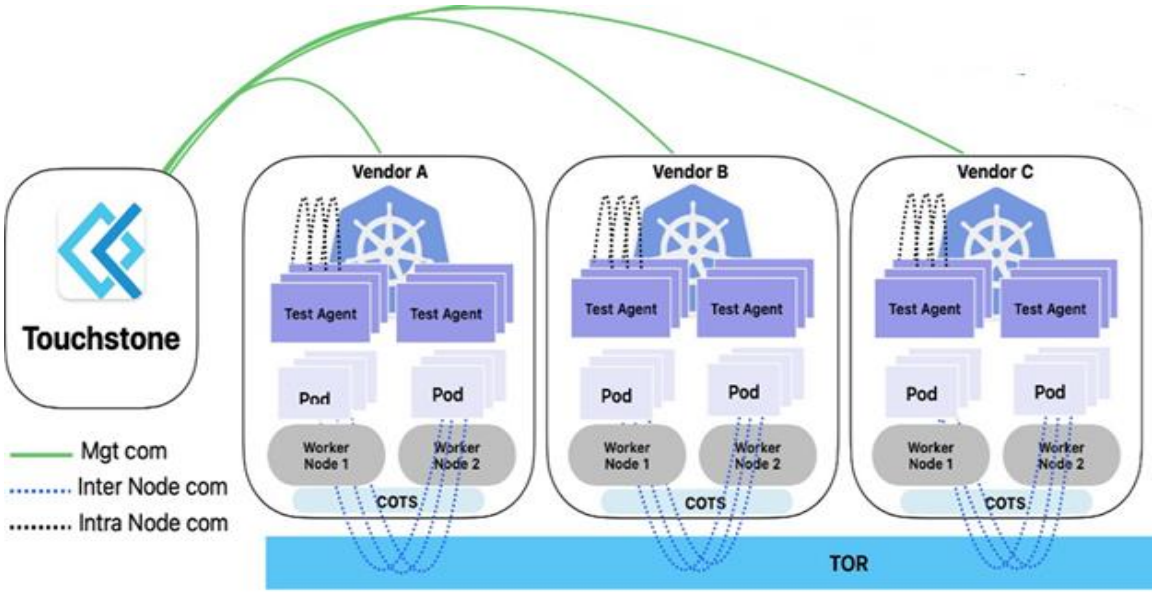


Figure 3 - VoerEir Touchstone Tool and Testing Cluster in Vodafone Lab

We carried out 500+ test cases in this benchmark, below figure presents the overall testing areas and number of cases per area. The primary focus areas of the tests were:

1. Kubernetes Functional Testing: Validation of Kubernetes features across different versions.
2. Kubernetes Security Configuration Testing: Adherence to the Kubernetes CIS Benchmark.
3. Kubernetes Control Plane (API) Performance: Responsiveness and scalability of the Kubernetes API Server load.
4. Kubernetes Performance: Compute, memory, storage and network performance
5. Robustness and Fault Tolerance: Recovery of the CaaS platform infrastructure plane in the event of component failure.
6. High Availability (HA): validate CaaS platform control plane high availability including Kubernetes API server

HA, Kubernetes master node HA, etcd HA etc.

7. Stability Test: Stability test over extended duration of 48 hours and under stress conditions.

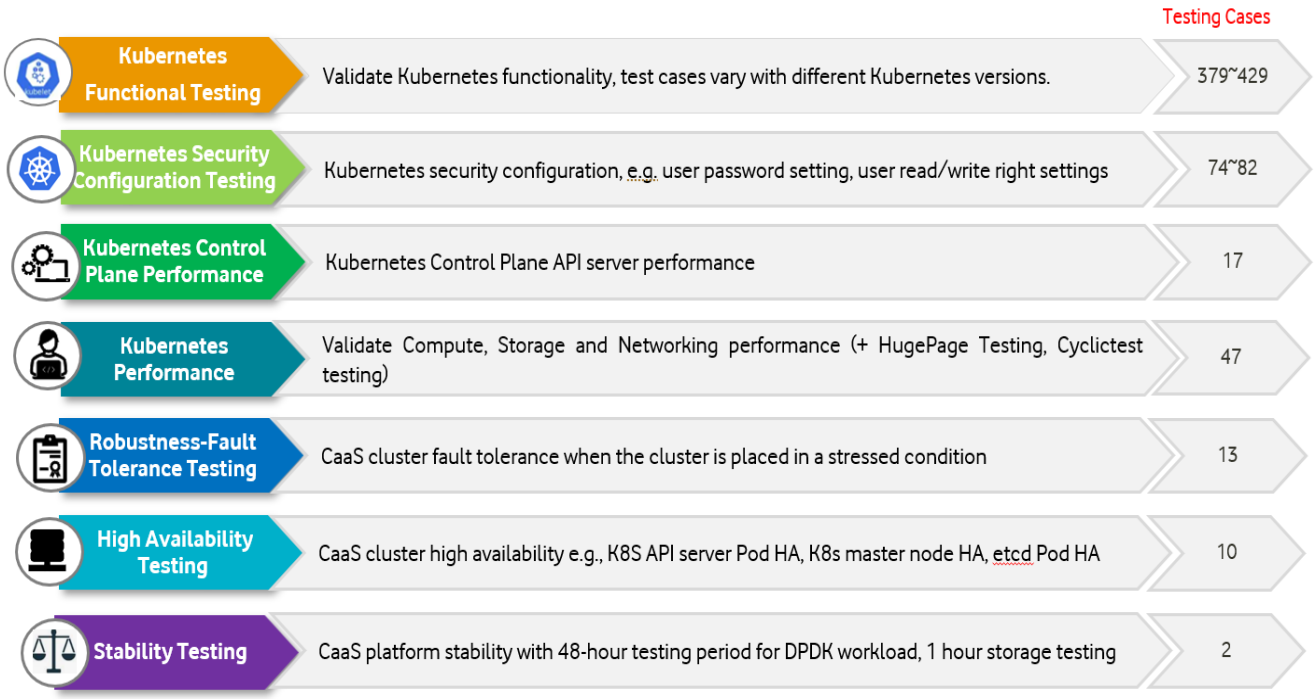


Figure 4 – O-Cloud Performance Benchmark Testing Areas and Testing Cases

The details of each testing areas are descripted in the following section.

1) Kubernetes Functional Testing

A suite of 379–429 test cases was executed to validate the functionality of Kubernetes across all vendors. These test cases checked various Kubernetes features, including pod scheduling, networking, storage, and RBAC (Roll Based Access Control) policies. The variation in the number of test cases across vendors was due to differences in the Kubernetes versions being used, with newer versions including additional features or changes in functionality.

2) Kubernetes Security Configuration Testing

This area will test conformance to the CIS (Center for Internet Security) security benchmark and reviews known vulnerabilities.

Around 74–82 test cases were conducted to validate the security compliance of the Kubernetes platform according to the Kubernetes CIS Benchmark. This testing evaluated various security aspects of the CaaS layer, such as configuration file ownership, permissions, network policies, and service security. The variance in the number of test cases was due to differences in Kubernetes versions, which included new security features and configurations.

3) Kubernetes Control Plane (API) Performance Testing

A critical component of Kubernetes performance is its API server's ability to handle concurrent requests efficiently. In this test, CRUD operations (Create, Read, Update, Delete) on various Kubernetes objects were executed concurrently to simulate real-world workloads. The API server's response time and throughput were measured under varying loads to assess its performance and scalability.

4) Kubernetes Performance Testing

A total of 47 test cases were executed to validate the core infrastructure performance. These tests were designed to verify how optimized the hardware and software configurations are to see if they meet performance targets.

Compute Performance

The purpose of this area is to detect underperforming Cluster Under Test that can directly lead to the need for more resources than needed (more cores. more memory. etc.).

- **Real Time Operating System (RT-OS) Latency**: The RT-OS latency was tested to measure how quickly the system can respond to scheduling real-time priority process requests, with a focus on minimizing overhead during heavy loads, that is, the time it takes for the real-time processes to be scheduled over non real-time processes that have lower priority.
- **Latency for Random Memory Access**: Measured the time taken for the system to access random memory locations. This ensures efficient memory management during high-demand scenarios.
- **Bandwidth of Random Memory Access**: The test validated the bandwidth of random memory access operation.
- **HugePage Memory Bandwidth**: This test focused on evaluating how efficiently large memory pages (HugePages) are utilized by the system to optimize memory access and performance.

Network Performance

Network performance covers network throughput, and network jitter & latency.

Network Throughput - This area will detect underperforming parts of the Networking stack provided by the Cluster Under Test.

Network Jitter & Latency - This area will detect latency problems in the Networking stack provided by the Cluster Under Test, which can lead to problems with the CNFs (Containerized Network Function) that use it and the real-time notification of synchronization problems.

Network performance was assessed in two primary scenarios: intra-node and inter-node communications.

- **Intra-node Performance**: In the intra-node setup, normal vNIC, Direct vNIC (SR-IOV - Single Root I/O Virtualization), and userspace DPDK were tested. Metrics such as latency, jitter, and throughput were measured to evaluate how effectively traffic was handled within a single node. These tests simulated real-world workloads to validate intra-node communication performance.
- **Inter-node Performance**: For inter-node communication, tests were conducted using normal vNIC, Direct vNIC (SR-IOV), and userspace DPDK. The focus was on measuring latency, jitter, and throughput across nodes to assess the system's ability to handle traffic between nodes under realistic workload conditions. The tests also evaluated the switching infrastructure's capability to support inter-node communication without introducing bottlenecks, packet loss or additional latency.

Storage Performance

Three storage performance scenarios were evaluated using fio on a Kubernetes pod to assess the system's ability to handle different I/O (Input/Output) workloads. These tests focused on measuring read/write latency and read/write IOPS (Input/Output Operations Per Second) across varying block sizes and queue depths to ensure the storage layer meets performance demands for containerized applications.

- Small Block Random I/O Performance: Tested 4kB block size, queue depth 16 to simulate workloads requiring frequent access to small data blocks, such as databases and logging services. The focus was on measuring IOPS and latency under high concurrency.
- Large Block Sequential/Random I/O Performance: Used 1MB block size, queue depth 1 to assess the system's ability to handle large block read/write operations, common in media processing and backup workloads. Read/write latency and IOPS were the key metrics.
- High-Concurrency Large Block I/O Performance: Increased concurrency with 1MB block size, queue depth 16 to evaluate how well the storage system handles multiple simultaneous I/O operations under heavy workloads.

5) Robustness Testing

A total of 13 test cases were designed to evaluate the robustness and auto-recovery capabilities of the CaaS platform under critical failure scenarios (Chaos testing). The key tests included:

- Control Plane Resilience: Introduced failures to vital control plane components, such as the Controller Manager, Scheduler, etcd (etcd is a distributed, key-value store), and API Server, to validate high availability and fault recovery mechanisms.
- Networking Pods/Service Recovery: Validated the ability of critical networking services, including CoreDNS (Domain Name System), Kube-Proxy, and CNI (Container Network Interface), to recover within an SLA timeframe. Assessed the platform's handling of network disruptions, including failures in these components, to ensure consistent and reliable connectivity.
- Self-Healing of Evicted Pods: Injected CPU, memory, and network stress to simulate pod evictions and test the platform's ability to automatically restore evicted pods. This experiment verified the system's self- healing capabilities, ensuring service continuity without manual intervention during resource-induced disruptions.

These tests were critical for validating the platform's resilience in recovering from real-world failures while maintaining high availability and minimizing service disruptions.

6) High Availability (HA) Testing

A total of 10 testing cases were designed to validate CaaS platform high availability and auto-recovery capabilities when CaaS platform was under critical failure situation. These testing cases cover Kubernetes API server HA, Kubernetes Controller Manager HA, Kube-scheduler HA, Core DNS HA, etcd HA, Kubelet service recover testing, Kubernetes Master Node HA, Kube Proxy HA, Default CNI HA.

7) Stability Testing

Two stability scenarios were used to evaluate the long-term reliability of the platform:

- Extended Stability Testing (48 hours): The platform was subjected to continuous stress over a 48-hour period to evaluate if any performance degradation occurred. This test ensured that the system remained stable under sustained loads.
- Storage Stability (1 hour): A more focused test on the storage layer was conducted over 1 hour, verifying if the storage subsystem could maintain high performance and low maximum latency during high load and stress.

4. Vodafone Lab Testing Environment and Testing Results

Vodafone built industry first CaaS performance benchmark lab environment in our Central Lab. This environment is designed for CaaS platform benchmark dedicatedly, and it enables us to run the benchmark activity in regular basis e.g. yearly testing.

4.1 Vodafone Lab Testing Environments

As shown in below figure, the lab environments consist of two components. The first component is "Touchstone" tool cluster which is deployed on 1 Dell R740 servers. The second component is CaaS cluster which is deployed on 2 identical COTS-servers for each supplier (refer to right hand side in below figure). To ensure consistency across the benchmarking exercise, each vendor was provided with dedicated hardware (e.g. 2 Dell XR8620t servers), configured identically. The performance tests for all vendors were conducted in parallel to achieve fair comparison except network performance, which was carried out sequentially to avoid the network congestion, thus avoid the impact on vendor results.

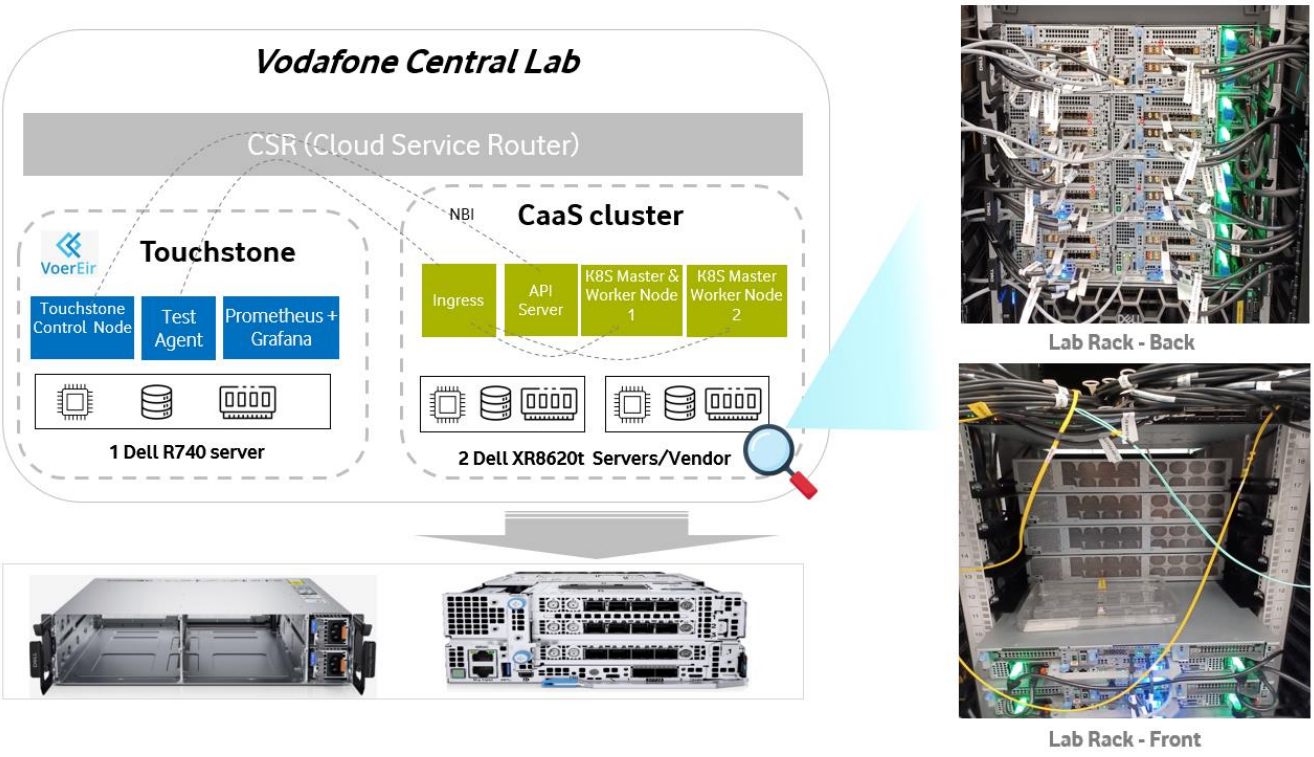


Figure 5- Vodafone O-Cloud Performance Benchmark Testing Environment

4.2 CaaS Performance Benchmark Testing Results

Vodafone constantly drives OpenRAN development via vendor benchmark process. We carry out summer and winter benchmark every year, and the benchmark covers End-to-End network components i.e. Radio, Transport, Core Network, Fix networks.

Beside feature and roadmap benchmark, Vodafone is driving true vendor performance benchmark based on actual measurements. For example, Vodafone built a Radio Access Network (RAN) network testing

environment with live sites deployed for Radio performance benchmarking, including traditional RAN and OpenRAN.

By building up a lab environment in our Central Lab, we expanded our benchmark capability into CaaS area. Within 2 months' validation, we delivered industry first O-Cloud (CaaS) performance benchmark inviting 3 tier-1 CaaS vendors. In this section, we'll present the CaaS performance benchmark results with vendors' names anonymized.

4.2.1 Compute Performance

Compute performance is an important KPI (Key Performance Indicator) to measure CaaS platform data processing capabilities. We validated below 4 individual testing cases in this category testing,

- **Real-Time Operating System (RT-OS) latency**: using Cyclictest tool to measure the maximum latency and average latency of scheduled tasks on the CPU. The OS latency was tested to measure the latency in real-time scheduling of requests, with a focus on minimizing overhead during heavy loads.
- **Bandwidth of Random Memory Access**: The test validated the bandwidth of random memory access operation.
- **Latency for Random Memory Access**: measure the time taken for CaaS system to access random memory locations.
- **HugePage Memory Throughput**: measure HugePage bandwidth.

Below figure illustrated average RT-OS latency and maximum RT-OS latency. All 3 vendors achieved average RT-OS latency less than 6µs, which meet the target SLA (10µs). But only vendor A and vendor C meet the maximum RT-OS target SLA (15µs). Vendor B failed to meet maximum target SLA due to their RT-OS kernel issues during the testing.

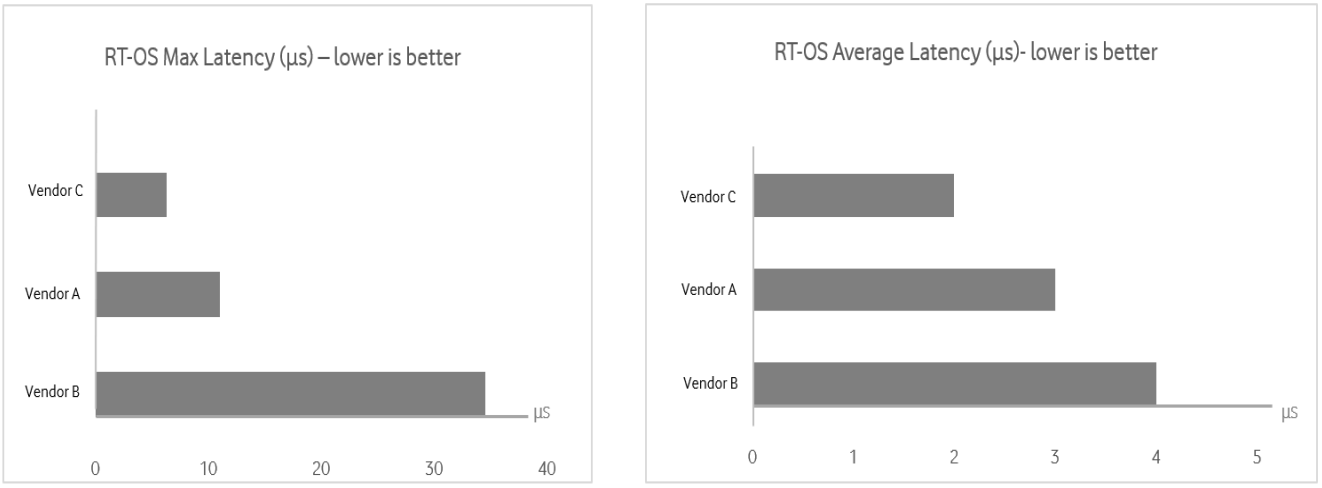


Figure 6- Real Time Operating System (RT-OS) Average and Maximum Latency Testing Results

HugePage memory throughput is a very important feature for RAN application as it reduces the impact of the page index management, provides minimal table look-aside times, and decreases memory access latency. It is also essential to support DPDK based line rate data processing for DU application.

Below figure showed HugePage memory throughput testing results. All three vendors meet the target SLA (7000MB/S).

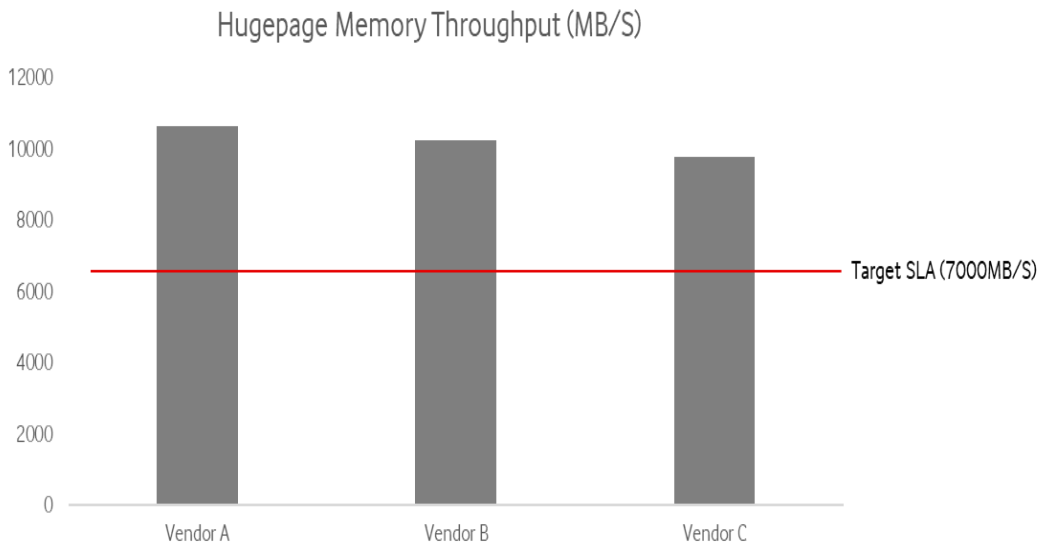


Figure 7 - HugePage Memory Throughput Testing Results

4.2.2 Network Performance (Throughput) Testing Results

Network Throughput testing is to measure Data Plane Network Throughput for intra-node and inter-node for below scenarios (shown in Figure 8):

- Normal vNIC throughput – using CaaS Kernel IP stack for non-time critical traffic
- Direct vNIC throughput – using Intel DPDK library for time critical traffic. This is a very important KPI for OpenRAN as it provides line rate throughput for data transmission between DU and CU interface

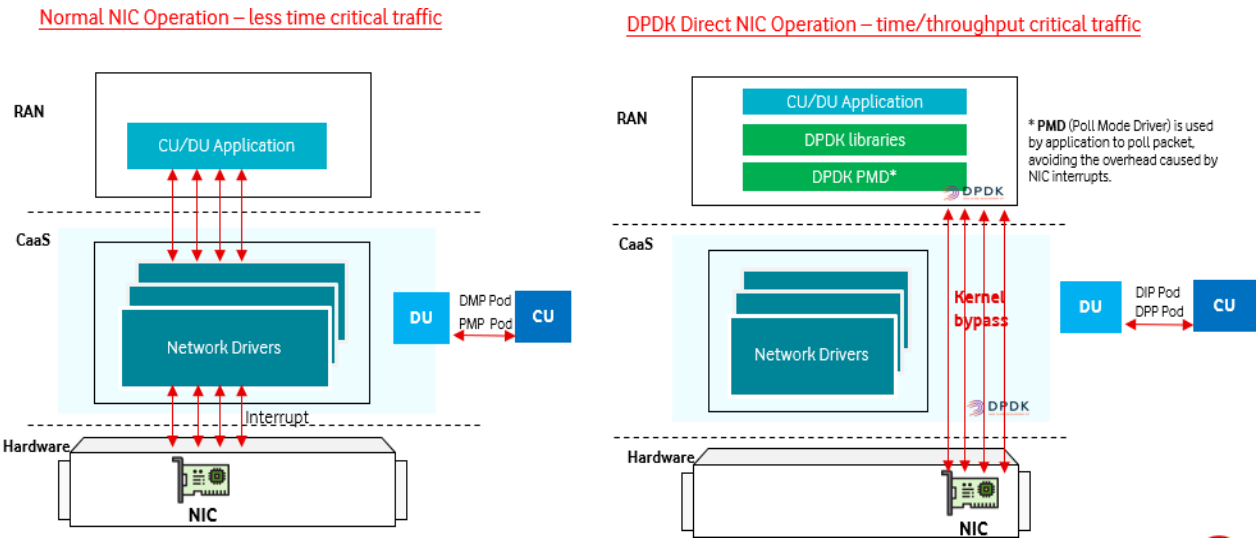


Figure 8- Normal NIC Operation and DPDK Operation

Below figure illustrated DPDK based intra-node and inter-node throughput. It is obvious that all 3 vendors achieved better throughputs than target SLA (35MPPS)

Figure 9 DPDK Intra-node and Inter-Node Throughput Testing Results

4.2.3 Network Performance (Jitter & Latency) Results

Network Jitter & Latency testing is to testing latency during data processing using CaaS network stack. It is another important KPI to measure the CaaS platform performance. Normally longer jitter and latency will result in package transmission delay and package drop, thus impact on RAN application.

Below figure demonstrated DPDK based maximum latency for intra-node and inter-node. All 3 vendors achieved the good results for intra-node latency and meet the target SLA (200µs). Vendor B and C passed the inter-node latency testing and meet the target SLA (200us), but vendor A has issue to achieve inter-node maximum latency.

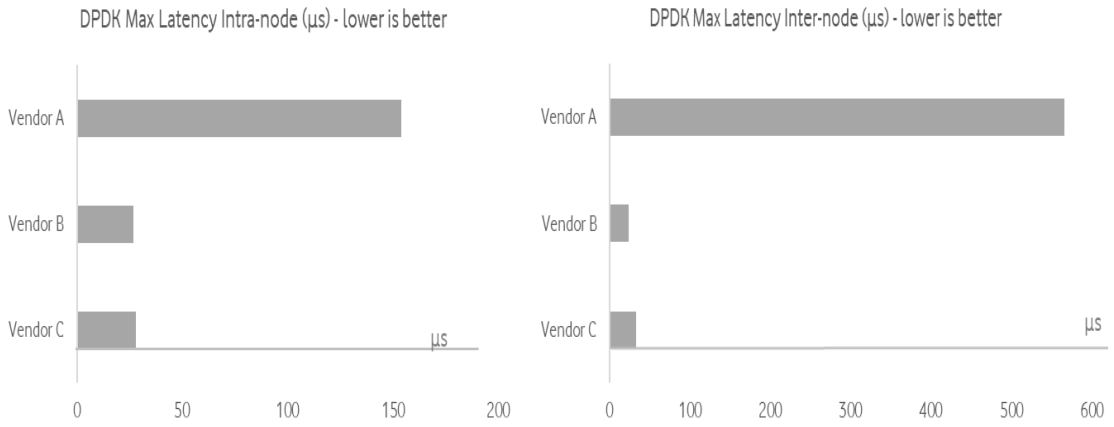


Figure 10 – DPDK Maximum Latency Intra-node and Inter-node Testing Results

4.2.4 Storage Performance Testing Results

Storage performance was tested on a Kubernetes pod to assess the system capability to handle different Input/Output (I/O) workloads. These testing focused on measuring Read/Write I/O latency using different block size and queue depths. For OpenRAN implementation, as CaaS layer shared the storage with RAN

application, and the storage for CaaS is only used for store Pod state and logs, thus it was configured as low priority in one of vendor configuration.

Below figure presented one of the storage testing results using big block request (1MB size). Among 3 vendors testing results, only vendor B met the target Read/Write IOPS SLA (1700), vendor A and C failed to meet the SLA. Vendor A configured storage as low priority considering OpenRAN implementation, and vendor C has issue with their storage configuration (e.g. without periodically housekeeping configured).

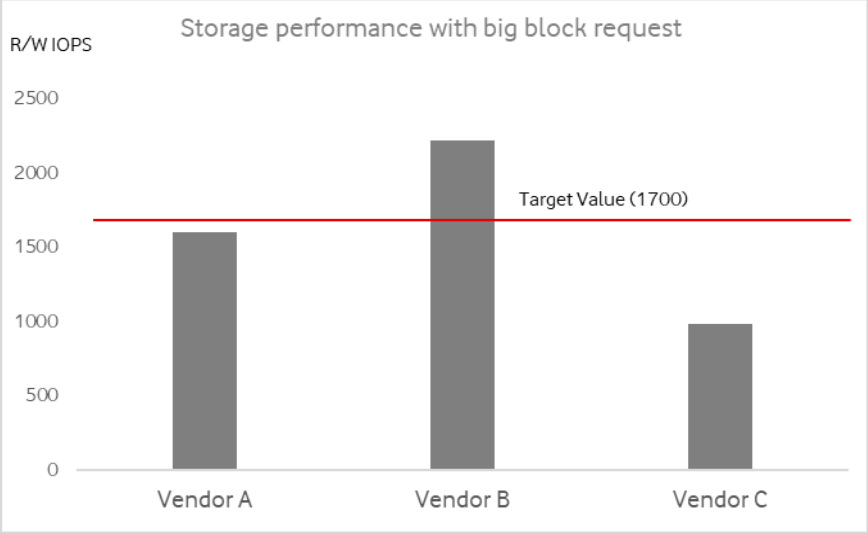


Figure 11 – Storage Performance Testing with Big Block Request

4.2.5 Kubernetes Functionality, Control Plane and Security Configuration Testing

In this Chapter, we'll provide the overall testing results on Kubernetes functionality, Control Plane and Security Conformance areas.

- Kubernetes Functionality Testing: Touchstone tool is designed to test Kubernetes functionality according to different Kubernetes version that vendor used. Thus, the test cases vary from 357 ~ 414. As shown in the figure below, 2 vendors passed Kubernetes functional testing, 1 vendor failed 5 cases (out of 414) due to DNS resolution failure. This means that all vendors did a good job of maintaining Kubernetes upstream compatibility while extending their platforms for manageability and observability

- Kubernetes Control Plane Testing: 17 testing cases were run to measure Kubernetes API server performance in terms of create read, update and delete (CRUD) operations for pod, namespaces, deployments and more. All three vendors passed the testing.

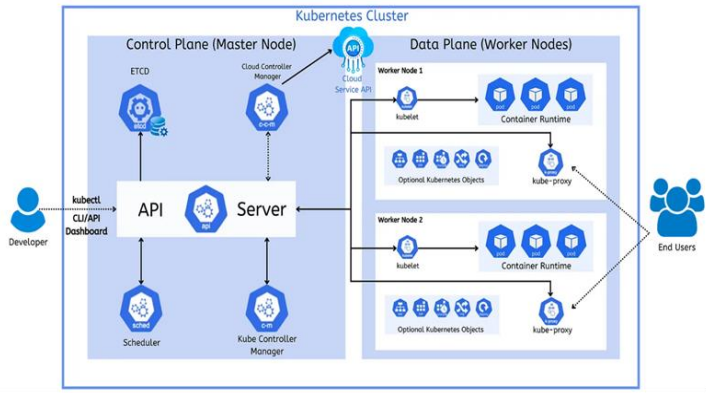
- Kubernetes Security Conformance Testing: these testing mainly to validate correct security configurations e.g. user read/write permissions, passwords etc. setting was corrected configured. During the testing, except vendor B, vendor A and C failed some of the testing cases, but the risks were all addressed by correcting the configuration after the results review session with vendors.

Kubernetes Function Testing

- All Kubernetes functional testing, vendor A and C passed and only vendor B failed 5 testing cases (among 414) due to DNS resolution failure.

Kubernetes Control Plane Performance

- Measures API server performance in fulfilling the API requests for create, read, update, delete (CRUD) operations.
- Testing comprised of CRUD requests for various Kubernetes resources like pods, namespaces, deployments, etc
- All 3 vendors passed all the 17 testing cases



Kubernetes Security Configuration Testing

- To validate whether correct security configuration (e.g. user read/write authority) are setting properly according to Kubernetes version
- Vendor B passed all the security configuration testing, Vendor A failed 3 testing case (out of 68) and Vendor C failed 3 testing cases (out of 77)
- All security findings were addressed during post-testing validation process

Figure 12 - Kubernetes Function, Control Plane and Security Configuration Testing

4.2.6 CaaS Platform Robustness and High Availability Testing

As CaaS platform is running cloud native OpenRAN application at edge RAN sites, thus platform resiliency and stability are essential to minimize service disruptions and guarantee best customer experience.

Below table summarize CaaS platform Robustness, High Availability testing results from all 3 vendors.

	Vendor A	Vendor B	Vendor C
Robustness Testing	12 passed, 1 failed due to node drain latency is above 60s due to HA configuration. Thus, the pod shut down gracefully.	Passed (13 out of 13)	Passed (7 passed, 6 skipped)
High Availability Testing	Passed (10 out of 10)	9 passed, 1 failed due to K8S master node high availability SLA= 300s, achieved 339s	9 passed, 1 failed on Master node reboot testing, down time SLA=300s, achieved 391s

Table 2 – CaaS Platform Robustness, High Availability Results

CaaS platform resiliency testing included Robustness testing and High Availability testing. These testing were critical for validating CaaS platform capability of self-recovery in stressful/failure conditions.

- Robustness Testing: It consists of 13 testing cases to cover CaaS control plane resiliency (e.g. Kubernetes master node, scheduler, etcd and API server), Network Pods/Service recovery (e.g. CoreDNS, Kube-Proxy and CNI) and Self-healing of evicted Pods. Vendor B and C passed the robustness testing, vendor A failed 1 testing case (1 out of 13) due to node drain latency is above 60s due to HA configurations, but very close to target KPI.

- High Availability Testing: 10 high availability testing were run to validate CaaS platform high availability in failure condition. Vendor A passed all testing cases, vendor B failed 1 testing case (1 out

of 10) due to Kubernetes master node HA latency is 339s, which very close to target SLA (300s). Vendor C failed 1 testing case (1 out of 10) due to Kubernetes master node rebooting latency 391s did not meet target SLA (300s), but very close to SLA.

5. Summary and Conclusions

5.1 Summary

This first benchmark demonstrates that the OpenRAN eco-system is enabled and supported with well performing and evolving CaaS platforms that are already supporting Global commercial deployments with several MNO's.

Overall performance summary and areas for improvement:

- Kubernetes Functionality and Security testing: All three vendors maintained great upstream compatibility with Kubernetes standard. For security conformance testing, we identified some of the security configuration mismatches in vendor's CaaS platform but were fully addressed during post-benchmark evaluation process.

- Network Performance: all three vendors achieved good results on Network Throughput and Jitter & Latency testing which demonstrated CaaS platform maturity to enable real-time OpenRAN DU/CU data processing. However, the testing indeed exposed configuration issues in DPDK setups in one specific vendor, emphasizing the need for precise and validated configurations to achieve desired performance levels.

- Robustness and High Availability: All vendors demonstrated significant resilience in robustness testing, highlighting the importance of a reliable control plane and network service recovery. Node drain latency and Kubernetes master node HA latency were critical metrics, with near-target KPIs observed.

- Storage Performance: Storage testing revealed that proper prioritization and periodic housekeeping are essential for meeting performance SLAs. The importance of efficient storage configurations was evident.

5.2 Conclusions

Overall O-Cloud is performing well across all vendors and ready to support the needs of OpenRAN workloads. This performance benchmark has provided essential feedback to participating O-Cloud vendors allowing them to focus their energy on key improvement areas for their roadmaps that ultimately enriches the O-Cloud capabilities available on the market.

Vodafone is proud to have delivered this first CaaS performance benchmark and share the findings openly across the ORAN industry. Not only delivering the industry first CaaS benchmark result but also establishing a Continuous Integration/Continuous Development (CI/CD) testing environment to continuously drive vendors' CaaS roadmap enhancements. Our systematic testing methodology, and identical testing environment for each vendor ensured fairness and accurate representation of vendor capabilities without interference, providing reliable benchmark results.

The benchmark results in this whitepaper highlighted varying degrees of compliance and capability across vendors. While all vendors showed strengths in certain areas, configuration issues underscored the importance of continuous optimization.

In conclusion, this first CaaS performance benchmark sets a precedent for ongoing evaluation and continuous improvement in OpenRAN implementations and establishes a baseline for further product evolution and development to bring further enhancement and efficiency to our OpenRAN platforms in areas such as test tools and methodologies, addressing configuration weaknesses, and driving performance and energy efficiency maintaining these comprehensive benchmarking activities and standards will be crucial to success.

Vodafone can offer a range of OpenRAN services to MNO's interested in making the transition to open Radio Networks, through technology and architecture, design, planning and benchmarking activities across all aspects of the platforms, Compute, Radio & SW.

If you would be interested to know more, please contact isaac.samuel@vodafone.com