SCIENCE CHINA
Information Sciences

• Supplementary File •

6G RAN Cloud Platform: A foundation for user-centric intelligent 6G

Nan LI[1,2], Xiaohua ZHANG[2], Qi SUN[2], Zhi WANG[2], Baichun ZHAO[2], Ting LI[2], Weichen NI[2], Jinri HUANG[2], Yuhong HUANG[2], Chih-Lin I[2], Mugen PENG[1] & Xiaoyun WANG[2*]

[1]*State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 1000876, China*
[2]*China Mobile Research Institute, Beijing 100053, China*

Appendix A User-centric transmission and sensing, resource management, and service provision

The limitations of the traditional cell-centric paradigm necessitate a fundamental shift towards User-Centric Networking (UCN). Unlike cell-based networks that attempt to optimize for system-level metrics, the UCN paradigm reorients the network's objective to delivering a guaranteed Quality of Experience (QoE) for each individual user, regardless of their location [1]. This is achieved by dynamically organizing network resources around the user, transcending rigid cell boundaries.

To realize this vision and transition from static management to a truly intelligent and adaptive network, an integrated framework is required, enabling continuous network perception, proactive decision-making, and the autonomous execution of high-level service goals. Thus the UCN hierarchical awareness framework is fundamentally realized across three hierarchical levels: transmission and sensing level, resource management level, and service provisioning level. Correspondingly, we proposed that to realize the user-centric feature at different level requires three key awareness capability: Spatial Awareness, Context Awareness, and Intent Awareness, as illustrated in Figure A1.

Appendix A.1 User-centric Spatial-aware transmission and coordinated sensing

The first and foundational awareness capability is Spatial Awareness, which is essential to realize the user-centric transmission and sensing. This capability enables the precise perception of users' physical location, velocity, and real-time channel characteristics of users across a wide area covered by multiple distributed Access Points (APs), as well as the surrounding environment. This foundational knowledge is critical to enable the dynamic and localized organization of network resources around the user.

Appendix A.1.1 *Cell-free architecture and coordinated transmission*

In a UCN, a large number of distributed APs jointly serve users through coordinated multipoint (CoMP) or cell-free massive MIMO. By enabling dynamic network resource organization around the user, the architecture inherently mitigates inter-cell interference and ensures high-quality signal reception, crucial for applications like immersive and high-reliability services that demand QoE uniformity [2]. This dynamic connectivity relies heavily on stringent requirements for coherence and synchronization among distributed APs for joint signal processing, such as joint precoding [3].

Appendix A.1.2 *Integrated sensing for user and environment perception*

Spatial Awareness is amplified by Coordinated Sensing, which fuses signals received from multiple distributed APs to achieve high-accuracy user positioning and velocity tracking. Integrated sensing and communication is a key enabler here, which leverages the same physical infrastructure and spectrum resources for both data transfer and sensing, enhancing spectral efficiency and promoting cost-effective hardware convergence [4]. The resulting real-time spatial data, such as user coordinates, velocity vectors, and predicted trajectories, is essential for the upper layers to manage resources. For instance, high-precision velocity tracking allows the network to proactively reserve computational resources or pre-stage network functions along a user's anticipated path, thereby preventing service degradation caused by sudden handovers and guaranteeing uniform QoE.

Appendix A.2 User-centric Context-aware resource management and optimization

Building upon the real-time spatial data feed, the second awareness capability is Context Awareness, which is essential to realize user-centric resource management. This capability empowers the fusion and analysis of multi-dimensional data streams, including user requirements and network conditions, to predict future states and behaviors. This predictive capability is key to enabling dynamic resource and function optimization, with the goal of satisfying individual user demands and optimizing personalized service performance. This proactive management leads to end-to-end dynamic scheduling.

* Corresponding author (email: wangxiaoyun@chinamobile.com)

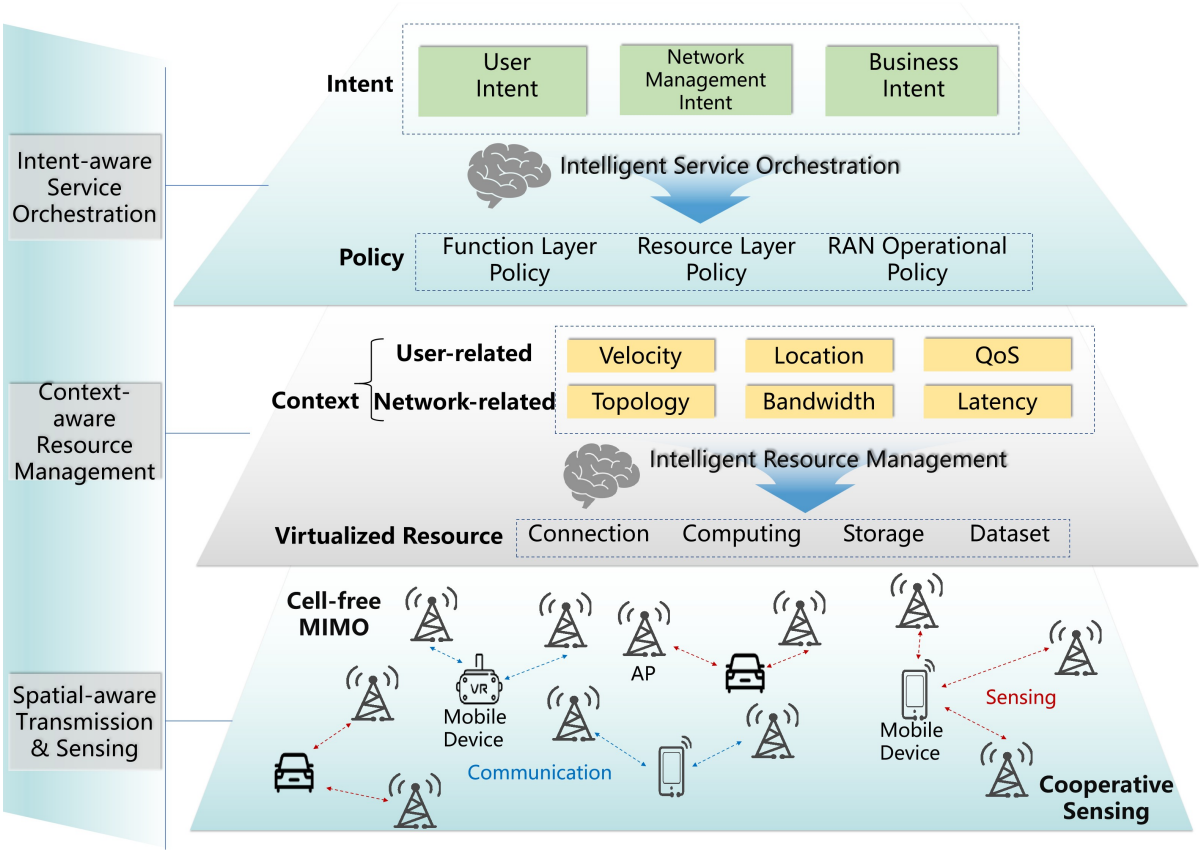

Figure A1 UCN hierarchical awareness framework

Appendix A.2.1 *Multi-dimensional context fusion and prediction*

Context information is typically categorized into two parts:

- **User and service context:** Includes dynamic QoE requirements, application type, mobility profile, device capability, and historical usage patterns.
- **Network and environment context:** Encompasses traffic load fluctuations, interference conditions, resource utilization, and operational status.

Data-driven Machine Learning (ML) algorithms are essential for fusing this disparate data, identifying correlations, and accurately predicting network and user behaviors (e.g., traffic peaks and user trajectories). These predictions serve as the basis for informed resource reservation and dynamic allocation strategies.

Appendix A.2.2 *Elastic resource and function provisioning*

Based on ML-driven decisions, the UCN executes dynamic adjustments at two levels:

- **Resource layer:** Resources such as spectrum, power, time slots, and bandwidth are dynamically allocated and re-allocated.
- **Function layer:** Driven by cloudification technologies like NFV, network functions can be dynamically instantiated, scaled, or migrated on demand.

This functional split and elasticity is crucial for matching computational capacity to the instantaneous demands of high-requirement users, thereby ensuring personalized service delivery and maximizing resource efficiency. This process is often managed through cross-layer optimization to achieve overall system efficiency while maintaining QoE targets.

Appendix A.3 User-centric Intent-aware service provisioning and automation

The highest and most transformative awareness capability is Intent Awareness, which is essential to realize the user-centric service provisioning. This capability translates high-level business goals or user requests directly into executable network policies and configurations. This mechanism is key to managing the complexity and scale of user-specific service customization.

Appendix A.3.1 *Intent modelling and abstraction*

Intent-aware networking allows users or vertical industries to express their objectives rather than specifying low-level operational details. Diverse intent inputs are processed through ML algorithms for modelling, parsing, and recognition. The result is a set of quantifiable, verifiable policies that guide the underlying network controls. This abstraction significantly lowers management complexity and enhances the usability and automation level of the network.

Appendix A.3.2 *Autonomous execution and closed-loop optimization*

The translated policies can then be executed within, e.g., a decentralized or multi-agent system framework, where individual intelligent agents collaborate to satisfy the global service intent while resolving potential policy conflicts. Coupled with real-time context perception, the network forms a closed loop: perception, decision, execution and optimization. This closed-loop mechanism ensures that user intentions—whether guaranteeing specific service demands for a single user or coordinating resources to meet cell-level KPIs—are continuously and consistently fulfilled in an autonomous manner.

Appendix B Mapping UCN visions to 6G RAN Cloud Platform requirements

The cloud platform could serve as the fundamental resource substrate and primary enabler for UCN. However, the paradigm shift toward UCN—driven by demanding 6G applications such as immersive XR and autonomous driving—imposes requirements on the cloud platform far beyond a traditional and static resource pool. These applications require that network functions dynamically adapt in real time to user mobility and context. To meet this challenge and natively support intelligent UCN, the cloud platform must evolve to enable three new hierarchical characters: (1) Spatial Awareness coordinated transmission and sensing; (2) Context Awareness dynamic and elastic resource management; and (3) Intent Awareness autonomous service provisioning. These three UCN characteristics translate into corresponding NF capabilities, which are further decomposed into interdependent cloud platform requirements for Infrastructure as a Service (IaaS) and Platform as a Service (PaaS) layers. This hierarchical mapping is illustrated in Figure B1, establishing the architectural foundation for a cloud-native platform capable of effectively supporting a user-centric, Artificial Intelligence (AI)-enabled 6G network.

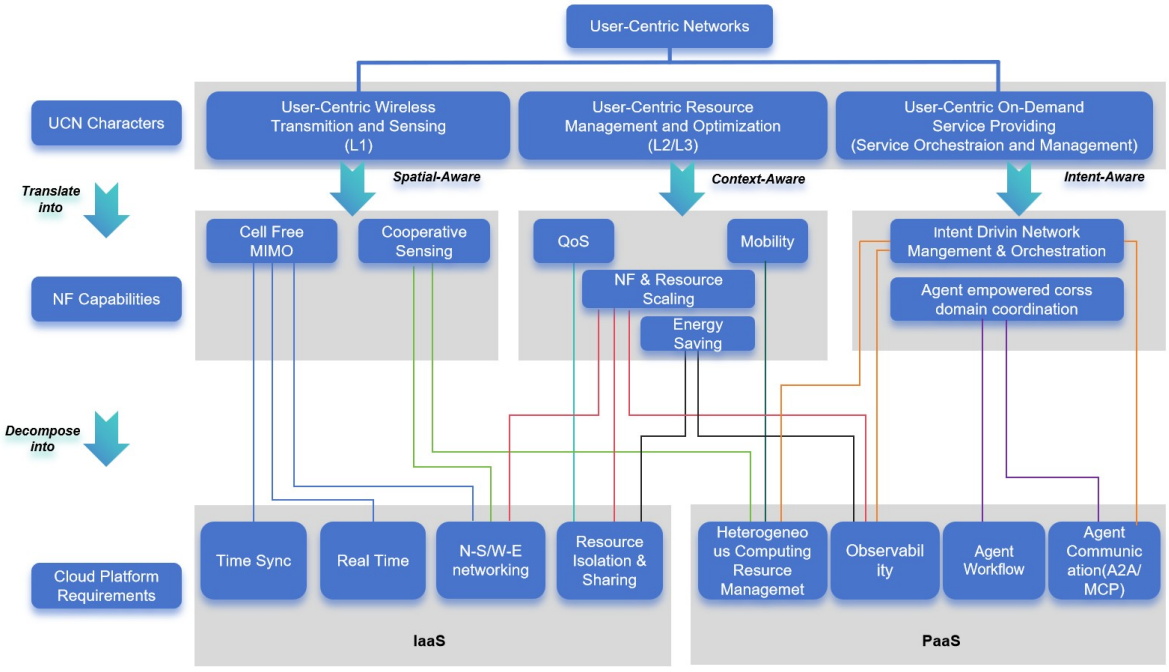

Figure B1 Mapping from UCN characters to 6G RAN Cloud Platform requirements

Appendix B.1 Support for UCN transmission and sensing (Spatial Awareness)

Spatial Awareness coordinated transmission and sensing encompass the essential functions of joint transmission and real-time environmental perception. To realize this capability, the cloud-native platform must meet a core, overarching requirement: enabling the coherent and real-time coordination of distributed radio and compute resources. This overarching requirement logically divides into two key areas. First, it demands a deterministic real-time foundation for IaaS to ensure coherence, encompassing (1) sub-microsecond synchronization to enable coherent joint transmission and sensing, and (2) deterministic ultra-low latency processing for real-time functions. Second, it requires high-performance data and workload processing to manage the coordination load, which necessitates (3) high-throughput networking on IaaS layer for massive data exchange between distributed units, and (4) latency-bounded computation on PaaS layer to execute AI-native sensing and channel prediction models within the RAN slot cycle. The following subsections will detail these specific platform requirements.

Appendix B.1.1 *Sub-microsecond synchronization*

In a User-Centric Network, Spatial Awareness is built on the instant ability of dozens of APs to act as a single, user-focused antenna array. The cloud platform must provide ultra-precise synchronization between distributed processing units (e.g., Distributed Units (DUs)) to enable coherent joint transmission and accurate sensing. The time synchronization precision must be $\leqslant 65$ ns, and the phase synchronization precision must be $\leqslant 2.5^\circ$ (equivalent to about 10 ns at 3.5 GHz) [5, 6].

This ensures timing misalignment is a small fraction (˜3 %) of a typical OFDM cyclic-prefix length for the 30 kHz sub-carrier spacing (3.2 us in 5G-NR normal-CP mode), preventing inter-symbol and inter-carrier interference. Phase coherence is critical for constructive signal combining in distributed MIMO systems. These values are explicitly specified in the O-RAN standards for multi-point coordination and are derived from the fundamental relationship between timing error and signal quality (Error Vector Magnitude) [5, 6].

Appendix B.1.2 *Deterministic ultra-low latency and real-time processing*

The platform must guarantee bounded, low-latency processing for real-time signal processing chains. The L1 user-plane processing latency must be within 200 microseconds for eMBB and 100 microseconds for uRLLC services. The one-way fronthaul latency (RU to DU) must be less than 150 microseconds. Consequently, the cloud platform must be designed to limit scheduling latency to under 10 microseconds, which is derived from practical engineering experience. This ensures sufficient time for maintaining the RAN stack state and processing data within the short slot duration.

These targets are derived from a bottom-up budget allocation of the 3GPP-defined 1 ms end-to-end air interface latency for uRLLC [7]. After accounting for radio frame duration (˜143 us), RF processing, and propagation delays, the residual budget for computation and transmission is severely constrained, necessitating sub-millisecond targets at each stage, as reflected in O-RAN and IEEE 1914.1 specifications [6, 8].

Appendix B.1.3 *High-throughput networking*

The 6G cloud-native RAN must move radio-rate traffic while meeting radio-time deadlines; both requirements stem directly from the physical-layer signal chain and the cell-free coordination loop, not from generic data-centre benchmarks. A 100 MHz, 64T64R O-RAN split-option 7-2x fronthaul carries 122.88 MS/second $\times$ 9 bit I/Q $\times$ 64 antennas = 24.3 Gb/s per RU—a value fixed by the eCPRI rate equation and independent of user-plane load [9]. The network must therefore supply 100 Gbps per server port with $< 1\%$ loss to avoid CRC-induced retransmissions that would break the 1 ms HARQ round-trip budget.

Appendix B.1.4 *AI-Native, latency-bounded computation*

UCN continually reshapes the "serving cluster" around each user. Every millisecond the set of access points that should jointly pre-code, schedule or hand over changes, and the radio parameters must be retuned per user, per slot. Rule-based optimisers cannot track this combinatorial churn at scale; AI becomes the only viable path to keep the network truly user-centric.

The platform must integrate AI accelerators, such as Graphics Processing Unit (GPU), Neural network Processing Unit (NPU), and support the execution of complex models within strict timing budgets. Inference latency for L1 AI models (e.g., for channel prediction) must be $\leqslant 50$ us. This 50 microseconds is ˜10% of a 5G-NR slot (0.5 milliseconds at 30 kHz), leaving enough time for the scheduler to use the AI result in the same slot. So that it can apply the prediction to the immediately following OFDM symbols without breaking the slot-level closed-loop control.

AI-based processing is replacing traditional algorithms in the physical layer but must adhere to the same latency budget. The scaling time ensures the network topology can dynamically adapt to changing user clusters without service degradation, a cornerstone of UCN elasticity.

Appendix B.2 Support for UCN resource management (Context Awareness)

By leveraging multi-dimensional context information provided by Spacial Awareness or intent-derived network policies, Context-aware resource management enables the network to transition from reactive responses to proactive, predictive optimization. By leveraging multi-dimensional context information, the UCN control plane can anticipate user demands and network fluctuations, dynamically adapting resources to optimize personalized service performance. To enable this predictive optimization, the cloud-native platform must meet a core, overarching requirement: enabling dynamic, stateful, and highly elastic resource management.

This overarching requirement logically divides into two key aspects. First, it demands agile resource control and provisioning, which encompasses (1) granular resource guarantees and isolation for IaaS and (2) dynamic computing resource managements well as observability for PaaS. Second, it must ensure seamless service continuity and state migration, which requires (3) support for stateful mobility, facilitated by (4) high-throughput intra-BBU networking for efficient state synchronization on IaaS layer. The following subsections will detail these specific platform requirements.

Appendix B.2.1 *Granular resource guarantees and isolation*

The UCN may trigger a serving-cluster update as often as every 200–500 ms. The earliest allowed instant is lower-bounded by the timeToTrigger that the network can configure—any value from 0–5.12 s in 40 ms steps [10]. We assume the operator sets it to 256–480 ms to balance measurement reliability and handover agility. And by the movement of a pedestrian at 3 km/h crossing the 0.3 m coherence distance in approximately 360 ms. Within this timeframe, the base station protocol stack must execute deterministic computations and time-critical tasks (e.g., channel estimation, precoding update). To ensure this level of determinism, every CPU cycle, cache line, and I/O operation must be quantitatively reserved and strictly isolated. Even small jitter—greater than a few microseconds—can disrupt beamforming weights and handover commands, making granular resource guarantees and isolation essential. Simultaneously, the system must continuously discover, measure, and report resource allocation—granted, consumed, and drifting—on a microsecond scale. This ensures that any violations are detected and corrected before the next cluster re-selection, driving the need for high-resolution observability.

- Fine-grained CPU stall, cache occupancy, memory-bandwidth and I/O latency histograms;
- Real-time anomaly detection (e.g., $> 5\%$ throughput drop) raised within 50 milliseconds;
- Open data model (OpenTelemetry / eBPF) so that external observability fabrics can correlate OS-level contention with 5G-core/O-RAN control-loop events.

Appendix B.2.2 *High-throughput intra-BBU networking*

In a UCN architecture like Cell-Free Massive MIMO, user data and channel state information need to be shared among a cluster of access points for joint processing. This generates a massive amount of east-west traffic within the BBU pool. To support intensive inter-base station collaboration (e.g., for CoMP), the cloud platform must provide high-bandwidth, low-latency east-west networking between BBU/DU pods. The bandwidth of the intra-cluster network must be scalable with supporting sustained data exchange rates of 100 Gbps between collaborating DU pairs [11].

Appendix B.2.3 *Stateful mobility and service continuity*

To support seamless user mobility in a cell-free architecture, the platform must enable rapid state migration. The transfer of a user's full session context (larger than 1 MB) between DUs must complete in 10 milliseconds [12].

This ensures that service interruption during handover is imperceptible and meets the stringent continuity requirements of advanced UCN services, as defined in 3GPP procedures for session and service continuity [12].

Appendix B.2.4 *Dynamic computing resource management*

The paradigm of UCN necessitates a fundamental shift from static resource allocation to dynamic computing resource management at the cloud-native platform level. This capability is the operational enabler for the core UCN characteristic of elastic function provisioning. Driven by cloud-native and Network Function Virtualization (NFV) technologies, network functions are no longer fixed entities but are required to be dynamically instantiated, scaled, or migrated in response to real-time user mobility and service demands. To support this agility, the underlying platform must provide fine-grained, flexible control over a distributed and heterogeneous resource pool, encompassing not only traditional CPUs but also specialized accelerators like GPUs. This involves the abstraction and pooling of these disparate resources, allowing them to be collectively managed and allocated on-demand. Consequently, dynamic computing resource management is not merely a feature but a foundational pillar, ensuring that the UCN can deliver on its promise of tailored performance while maintaining operational and energy efficiency.

Appendix B.3 Support for UCN service provisioning (Intent Awareness)

Intent Awareness serves as the highest level of network automation, translating high-level user goals or business objectives directly into executable network policies and configurations. This allows the network to autonomously handle dynamic service requests, adjusting to diverse and changing user demands without manual intervention. To translate this vision into practice, the cloud platform must provide an autonomous, AI-driven orchestration framework capable of converting abstract goals into executable, real-time network actions.

This capability is built upon two essential pillars for PaaS layer. The first is the execution engine, demanding the ability to implement translated policies with extreme speed and reliability, enabling rapid and responsive service instantiation. This requires 6G RAN Cloud platform enhancing (1) real-time resource availability monitoring and (2) dynamic resource scheduling. The second is the coordination fabric, requiring a robust and scalable system for AI agents to manage these actions, which is achieved through an (3) agent workflow and (4) agent gateway. The specific requirements for these two pillars are explored below.

Appendix B.3.1 *Rapid and responsive service instantiation*

The orchestration system must efficiently translate high-level declarative user intents (e.g., for a specific network slice or service) into a fully provisioned, operational, and cross-domain service spanning RAN, Transport, Core, and Edge networks. This process requires the system to coordinate multiple subsystems and ensure seamless resource allocation across these domains in real time. Given the dynamic and user-specific nature of 6G services, this orchestration must occur within stringent time constraints—no more than 60 seconds, as specified by industry standards [13]. To achieve this, the platform must integrate intelligent automation, real-time resource availability assessment, and fast provisioning mechanisms across different network layers. Additionally, the system must handle dynamic adjustments, such as resource conflicts and network fluctuations, ensuring that the service remains fully operational from the moment it is instantiated, with continuous monitoring and optimization. This level of responsiveness is critical for providing personalized services with minimal delay, supporting the ultra-low latency demands of 6G networks.

Appendix B.3.2 *Agent workflow and gateway*

In user-centric networking, the only fixed element is the user's intent; everything else—cells, bandwidth, compute—must be continuously rewritten as the user moves, changes the application, or simply changes their mind. Natural-language intent therefore becomes the primary control signal. A lightweight Large-Language Model (LLM), embedded in the cluster and trained in network telemetry and service logs, converts free-form requests into executable workflows whose steps carry deadlines aligned to radio slots and coherence windows. An agent gateway fronts every domain (radio, transport, core, edge) and exposes a single, secure call surface; each workflow vertex is translated into the appropriate south-bound command—retune antennas, scale a function, steer traffic, claim an accelerator—while the gateway streams results back to the agent in real time. Agent-to-agent calls and shared-context tokens move the user's original goal across protocols and connections without restart or rediscovery. Thus Intent Awareness, LLM-based interpretation, agent workflow, and gateway-based tool invocation form an inseparable triad: the network understands what the user wants, negotiates resources domain-by-domain, and keeps the service pinned to the user rather than to a fixed cell.

Appendix C 6G RAN Cloud Platform architecture

The functional and performance requirements mandated by the UCN delineate a clear set of quantitative, cloud-side obligations. These obligations, derived from the three foundational pillars of Spatial, Context, and Intent Awareness, cannot be fulfilled by generic cloud

architectures. To bridge this critical implementation gap and provide a practical foundation for deploying UCN, this section introduces the 6G RAN Cloud Platform architecture, presented in Figure C1. The reference architecture is structured into two primary layers: the IaaS Layer and the PaaS layer. The IaaS layer offers fundamental infrastructure resources such as computing, storage, and networking, forming the foundation for deploying and operating network functions. The PaaS layer, built on top of IaaS, provides advanced capabilities and services, including development and runtime environments for applications. Together, these layers are designed to natively support the hierarchical awareness model of UCN, enabling efficient resource utilization and facilitating the rapid development of intelligent, user-centric applications.

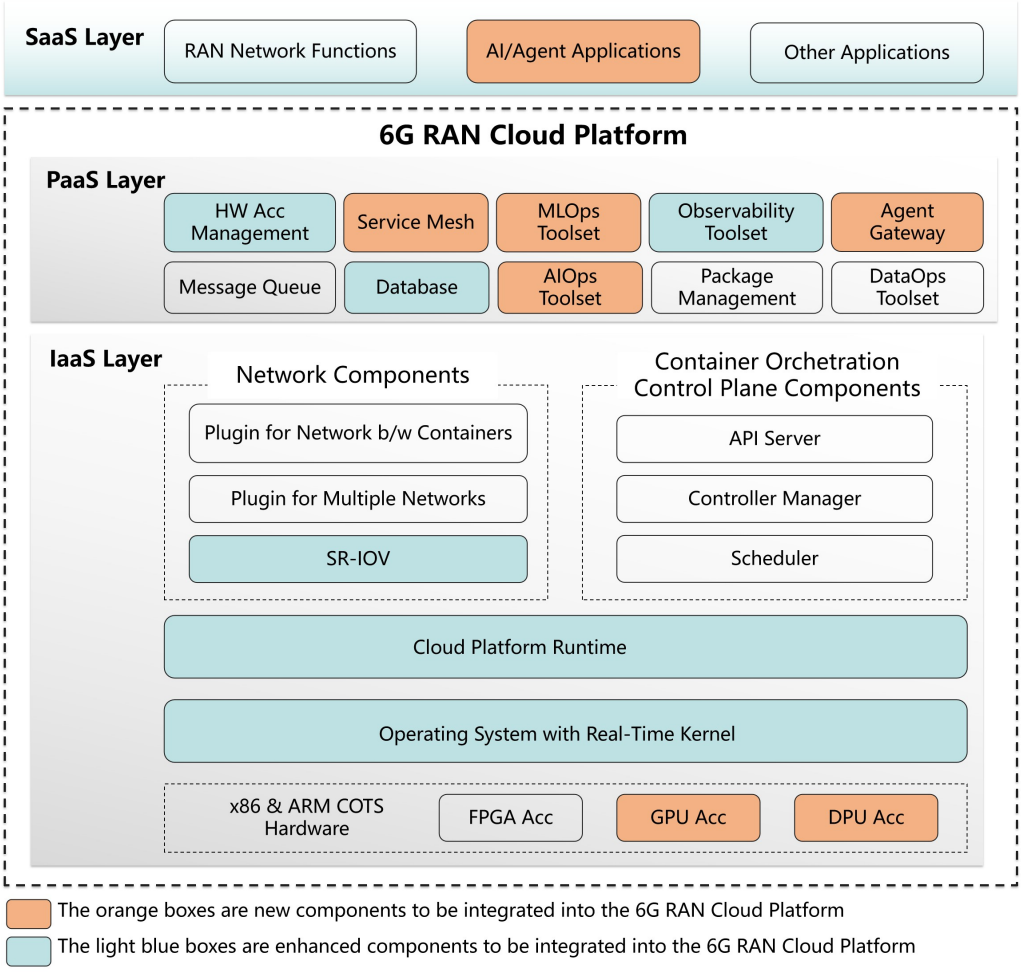

Figure C1 6G RAN Cloud Platform architecture for UCN

As illustrated in Figure C1, components enhanced beyond the O-RAN O-Cloud baseline are highlighted in light blue, while entirely new capabilities introduced for 6G RAN Cloud Platform are marked in orange. Table C1 systematically contrasts the two platforms across IaaS and PaaS layers. At the IaaS layer, our platform enforces deterministic sub-10 µs scheduling, 100 Gbps fronthaul with $<1\%$ loss, and sub-microsecond synchronization ($\leqslant$65 ns) to meet UCN's spatial-awareness demands. At the PaaS layer, we extend O-RAN with unified heterogeneous accelerator management (including Data Processing Unit (DPU) and Multi-Instance GPU (MIG)), AI-driven observability for predictive auto-scaling, and native data services beyond basic etcd. More fundamentally, MLOps toolset, service mesh, and agent gateway—all absent in O-RAN specifications—are introduced as first-class platform services, enabling automation and intent-driven UCN interaction. Together, these enhancements transform the cloud infrastructure from a virtualized RAN host into an intelligent, adaptive fabric purpose-built for 6G User-Centric Networking.

Appendix C.1 IaaS layer of 6G RAN Cloud Platform

The IaaS layer of the 6G RAN Cloud Platform provides essential infrastructure resources that are crucial for the deployment and operation of 6G network functions and applications. It comprises a set of hardware and software components. The hardware includes compute, network, and storage components and heterogeneous hardware accelerators, such as Field Programmable Gate Array (FPGA), GPU, and DPU. The computation-intensive or timing-intensive tasks of the radio physical layer processing, radio protocol stack, and the AI-related processing can be offloaded to the appropriate accelerators to improve the computing efficiency and energy efficiency. The software part of the IaaS layer generally includes real-time Operating System (OS), container runtime, network components, and container orchestration components.

Table C1 Comparison of O-RAN O-Cloud and 6G RAN Cloud Platform architectures

Layer	Feature	O-RAN O-Cloud [14]	6G RAN Cloud Platform
IaaS	Real-time Operating System	Max interrupt latency $\leqslant 20\,\mu$s (cyclictest)	Deterministic scheduling $<$**10** μ**s**, worst-case delay considered
	Cloud platform runtime	1) Environment for containerized workloads 2) PTP (G.8275.1/G.8275.2)	1) Same as O-RAN 2) **Sub-microsecond sync ($\leqslant$65 ns precision)** for UCN Spatial-aware transmission/sensing
	Network components	1) SR-IOV, OVS+DPDK 2) kube-proxy + CNI (Multus, SR-IOV)	1) Same as O-RAN 2) **100 Gbps/server, $<$1% packet loss**
	Container orchestration control plane	Kubernetes control plane (API server, controller manager, scheduler)	Same as O-RAN (tightly integrated with CNI/accelerator plugins for UCN service migration)
PaaS	Database	Basic etcd for Kubernetes	Native PaaS service: scalable, reliable storage (config, metrics, user data, AI artifacts)
	Software management	SW updates, rolling upgrade, VM/container migration	Same as O-RAN
	Hardware accelerator management	Abstraction for FPGA, GPU, DSP, ASIC (RAN offload)	Unified management for RAN and AI accelerators; **AI accelerators technologies, such as MIG**
	Observability toolset	Performance, alarm, log management for O-Cloud and RAN workloads	**AI/ML integration (e.g., GenAI)** for load prediction, energy-saving auto-scaling
	Service mesh	*N/A*	**Native communication management for RAN microservices and edge AI applications**
	MLOps toolset	*N/A*	**Full MLOps lifecycle**: data pipelines, model training, versioning, A/B testing, online inference
	Agent gateway	*N/A*	**Native support for Agent-to-Agent Protocol (A2A) and Model Context Protocol (MCP)**

Appendix C.1.1 *Real-time Operating System*

As analysed in Section Appendix B, the 6G RAN Cloud Platform must ensure that CPU scheduling latency for critical tasks, such as RAN data processing, remains under 10 microseconds to satisfy the demanding requirements of 6G RAN. To accomplish this, the platform employs an OS equipped with a real-time kernel. In addition, the system incorporates a preemptive scheduling kernel, CPU isolation, IRQ (Interrupt Request) affinity, NOHZ (No Hertz) mode, CPU pinning technology, etc. These combinations work together to guarantee the deterministic scheduling latency performance necessary for the platform's operations. However, it should be noted that the real-time performance of the system may be subject to cumulative effects over certain periods of operation. Specifically, there may be a moment or short period in which the system must address accumulated routine tasks; this could result in a temporary increase in scheduling latency. This temporary increase can compromise the system's ability to execute critical tasks in a timely manner, which is particularly concerning for the proper functioning of the platform. For example, when designing the software for a base station especially for a DU, it is imperative to take into account the maximum scheduling latency. This parameter serves as a crucial reference for establishing the timing and logic of the design, ensuring that the system can maintain its performance standards even when faced with cumulative effects that may arise during prolonged operation [15, 16].

Appendix C.1.2 *Container runtime*

It provides the necessary environment to run containerized workloads, facilitating the efficient and scalable deployment of network functions and applications. Compared to traditional virtual machine technology, containers are lightweight and helpful in reducing the operational cost [17].

Appendix C.1.3 *Network components*

To meet the high-performance networking requirements in Section Appendix B, network components such as Open vSwitch (OVS)/Data Plane Development Kit (DPDK) [18], Single Root I/O Virtualization (SR-IOV), plugin for network between containers, and plugin for multiple networks are introduced into the 6G RAN Cloud Platform architecture.

1. OVS+DPDK: The integration of OVS with the DPDK constitutes an advanced solution tailored for RAN environments where high-performance networking is paramount. The principal advantages of this integration are as follows:

   • Enhanced data processing velocity: DPDK facilitates the circumvention of the conventional kernel networking stack, empowering OVS to process data packets at elevated velocities. This capability is indispensable for meeting the stringent low-latency demands inherent to RAN operations.

   • Elevated scalability: The OVS+DPDK architecture is adept at managing the escalating data volumes within RAN, especially as the integration of AI functionalities—such as real-time traffic analysis and predictive maintenance—becomes more prevalent.

   • Optimized virtualization: This solution underpins the virtualization of network functions, offering the agility to deploy applications independently of the underlying hardware specifics.

2. SR-IOV: SR-IOV introduces a technique for the direct allocation of virtual network interfaces to virtual machines or containers, a feature particularly advantageous in RAN settings where unimpeded access to hardware resources substantially enhances performance. The core benefits of SR-IOV are:

   • Unmediated hardware interface: SR-IOV empowers virtual functions with direct access to the physical Network Interface Card (NIC), thereby diminishing latency and CPU overhead that are typically associated with hypervisor-based virtualization methods.

   • Ensuring isolation: It delivers hardware-level isolation among virtual functions, a critical aspect for preserving service quality within multi-tenant RAN environments.

   • Streamlined traffic management: SR-IOV's ability to bypass the hypervisor streamlines traffic management within RAN, a feature particularly advantageous for AI applications necessitating immediate access to network data.

While SR-IOV and OVS-DPDK deliver the raw throughput and latency performance required by the RAN data plane, their benefits can only be fully exploited if the container ecosystem is able to dynamically attach Pods to the correct accelerators, enforce per-tenant policies, and re-wire connectivity as the UCN "serving cluster" migrates across the pool. This control-plane agility is provided by a set of Kubernetes CNI plugins that sit logically above the two data-plane mechanisms and orchestrate their use at Pod life-cycle time. The plugin for networking between containers plays a crucial role in enhancing the performance and scalability of network communication within a cluster. It provides essential functionalities such as network policy implementation and IP address management, ensuring efficient communication between Pods. Moreover, it supports cross-node Pod networking, enabling seamless interaction across different nodes in the cluster. Simultaneously, the plugin offers network-level security isolation, which is vital for supporting multi-tenant deployments by ensuring the security and isolation of each tenant's network environment. Additionally, the plugin for multiple networks provides extended capabilities, allowing a single Pod to be attached to multiple networks and thereby increasing the flexibility and adaptability of network configurations to meet diverse application requirements.

Appendix C.1.4 *Container orchestration control plane components*

Container Orchestration includes the deployment, scaling and operation of containers. It ensures efficient and reliable management of containerized 6G RAN function workload. Key elements of the control plane include the Application Programming Interface (API) server, controller manager and scheduler. Kubernetes control plane is leveraged as the reference design [19].

Appendix C.2 PaaS layer of 6G RAN Cloud Platform

The PaaS layer of the 6G RAN Cloud Platform sits between the IaaS layer and the containerized RAN functionalities and applications. It generally encompasses a range of tools and services designed to facilitate the development, deployment, and management of applications in a cloud environment. It allows developers to develop, run, and manage applications without dealing with the complexity of building and maintaining the underlying infrastructure. As shown in Figure C1, the main component of the PaaS layer includes Database, Message queue, Package management, Observability toolset, Hardware accelerator management, Service mesh, MLOps toolset.

• Database provides scalable and reliable storage solutions for various types of data, including configuration data, performance metrics, and user-related information.

• Message queue facilitates asynchronous communication, enabling decoupling between producers and consumers of data and ensuring efficient coordination and synchronization of RAN-related tasks and processes.

• Software management supports deploying, upgrading, rolling update and deleting the software package across all nodes of the RAN cloud.

• Hardware accelerator management manages the hardware accelerators and support mapping them to RAN function containers and edge AI application containers. It provides heterogeneous compute awareness, scheduling and unified abstraction of multi-vendor accelerators (e.g., FPGA, GPU, DPU), enabling seamless resource scheduling and service continuity across diverse hardware platforms.

• Observability toolset provides tools for monitoring RAN application performance, collecting logs/alarms/faults and generating insights to aid in troubleshooting and optimization. For example, RAN traffic load and computation load of the cloud platform can be monitored and predicted for proactive scaling up and down for energy saving. The Observability tools can be further integrated with the advanced AI/ML technology, e.g., generative AI, to further improve the network automation level [20].

• Service mesh manages communication between microservices of RAN functions and edge AI applications.

• MLOps toolset integrates machine learning workflows into the network operations and facilitates the development, deployment, and management of ML models within RAN and for edge applications.

• Agent gateway with Agent-to-Agent Protocol (A2A) and Model Context Protocol (MCP) support, Agent gateway brokers every agent handshake, translating A2A identities and MCP context into secure, policy-governed calls so that any model can discover, invoke and bill another without new code while audit trails, rate limits and cost caps run silently underneath [21, 22].

Appendix D Experimental validation

Appendix D.1 6G RAN Cloud Platform testbed implementation

To validate that the proposed 6G RAN Cloud Platform can serve as a practical foundation for UCN by meeting its stringent requirements, we have developed an open 6G Cloud RAN Platform testbed that features flexible setups for various deployment scales, heterogeneous processing units such as CPUs, GPUs, DPUs, and FPGAs to handle diverse workloads, and it integrates open-source components such as Kubernetes and other tools for efficient cloud design. This testbed is essential for 6G technology testing as it evaluates both communication and computation capabilities. By leveraging IaaS and PaaS functions to support BBU, AI and other workloads, it enables the fulfillment of User-Centric Network applications. The overall structure and settings of the testbed are shown in Figure D1.

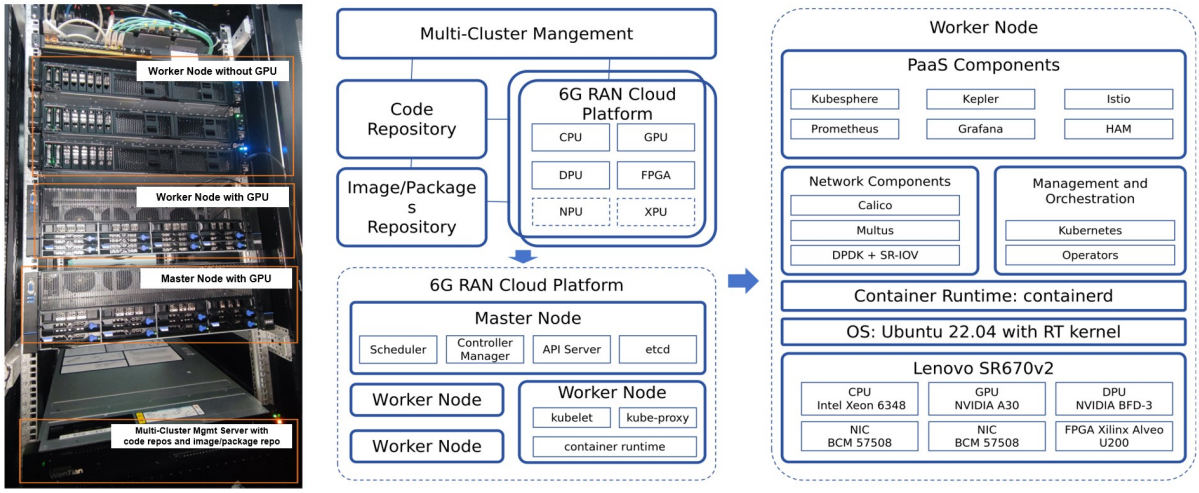

Figure D1 6G RAN Cloud Platform testbed settings

The detailed information of key components in the 6G RAN Cloud Platform is listed in Table D1. The selection of components follows the principle of adopting mature, widely-supported open-source software that aligns with the functional and performance requirements of UCN. These components provide the necessary APIs for RAN research—such as Kubernetes CNI interfaces, the NVIDIA GPU operator, and Calico CRDs—enabling fine-grained resource orchestration and hardware acceleration. Although the versions deployed are intentionally behind the latest releases, this strategy prioritizes proven reliability and the availability of upstream hotfixes over cutting-edge features. Each component has undergone multiple cycles of community hardening, providing a well-debugged and stable foundation. And these components have been long-term and rigorously validated through extended experimental testing, confirming their high reliability and ensuring easy reproducibility and migration. Consequently, the testbed is well-suited to support 6G RAN-in-the-cloud experimentation and the development of UCN applications.

Table D1 List of components in the testbed

Components	Details
Server	Lenovo SR670v2 with two CPU sockets
CPU	Intel Xeon Gold 6348 @2.6GHz, 28 cores
GPU	NVIDIA A30 24GB
FPGA	Xilinx Alveo U200
DPU	NVIDIA Bluefield-3
NIC	Broadcom 57508, Intel E810
Operating System	Ubuntu 22.04
Real-time kernel	5.15.119-rt
Container runtime	Containerd 1.7.5
Container orchestration	Kubernetes 1.26.8
Plugin for network b/w Containers	Calico 3.26.1
Plugin for multiple network	Multus 3.8
Observability toolset	Prometheus 2.39.1
Power consumption evaluation	Kepler 0.7.8

Appendix D.2 Validation methodology and results

To validate that the 6G RAN Cloud Platform can indeed serve as a practical enabler for UCN, we conducted a comprehensive evaluation on the open testbed, focusing on four key dimensions that directly map to the UCN awareness pillars introduced in Section Appendix B. Specifically, we assessed:

• Real-time performance test, corresponds to the deterministic ultra-low latency and real-time processing requirement of Spatial Awareness, particularly focusing on reducing task scheduling latency. This is critical for maintaining the RAN stack state and processing data within the short slot duration to achieve seamless joint transmission in UCN.

• Clock synchronization test, aligns with the sub-microsecond synchronization component of Spatial Awareness, which is essential for multi-access point coordination and sensing.

• Container network performance test, verifies the high-throughput for both physical layer of Spatial Awareness, and intra-BBU networking requirement under Context Awareness, ensuring efficient data exchange in dynamic UCN topologies.

• AI capability validation and GPU power consumption test, addresses the dynamic computing resource management for AI-intensive tasks in Context Awareness, highlighting the platform's ability to manage energy consumption and resource allocation efficiently.

These evaluations are not isolated benchmarks; rather, they are designed to confirm that the platform can simultaneously meet the stringent performance, intelligence, and autonomy requirements imposed by the UCN paradigm. By demonstrating deterministic real-time behavior, high-throughput networking, AI-ready infrastructure, and energy-driven scheduling, the testbed provides empirical evidence that the 6G RAN Cloud Platform is capable of translating the UCN vision—where network resources are continuously reshaped around the user—into a deployable, reproducible reality.

Appendix D.2.1 *Real-time performance test*

In UCN, real-time signal processing chains, such as those for distributed MIMO and coordinated beamforming, require bounded and minimal task scheduling latency to ensure seamless service delivery. In this test, the core metric evaluated is the system interrupt scheduling latency, defined as the time interval from when a hardware interrupt is generated until its corresponding service routine is scheduled for execution. This latency is a direct measure of the kernel's real-time responsiveness. A function generating periodic interrupts was deployed.

The test contains 11 test cases, each case tests more than 1.38 billion interrupts in a 48-hour period continuously, with different CPU load stress, which can be regarded as background workload, varies from 0 percent CPU usage to 100 percent CPU usage with interval of 10 percent. In each test case, the two primary metrics of interest are the average and maximum interrupt scheduling latency, as they provide insight into the kernel's real-time responsiveness and reliability [23]. The 48-hour period testing, with long-term cyclictest evaluations are essential for detecting cumulative CPU scheduling delays that may arise from mechanisms such as Read-Copy Update (RCU), potentially disrupting real-time performance. The significance of these evaluations is as follows:

• **RCU grace periods**: RCU is a mechanism designed to ensure safe memory updates in a multi-threaded environment. However, it can introduce delays during its grace periods, which are intervals required for readers to finish before a new data structure version can be made visible. These delays can accumulate over time, thereby increasing scheduling latency.

• **Resource contention**: Long-term testing can reveal the dynamics of how system resources, including memory, input/output (I/O), and CPU, compete with each other and influence one another over an extended period.

• **Thermal and power effects**: It provides insights into how CPU throttling due to temperature considerations and power-saving modes can affect scheduling performance.

• **Long-term stability**: It assesses the system's ability to sustain consistent real-time performance, identifying whether issues such as memory leaks or hardware degradation could lead to performance degradation over time.

Over 1.38 billion interrupts were sampled per test case. The testing results for scheduling latency distributions under varying CPU load conditions are shown in Figure D2, which indicates the number of occurrences within latency ranges from 2 microseconds to 10 microseconds by various colors. This observes us how the frequency of events within these latency ranges changes as the CPU load varies. And the statistical properties are further summarized in Table D2, which includes median, mean and maximum scheduling latency for CPU load percentages, spanning from 0% to 100%. This help us understand the kernel's real-time responsiveness and reliability.

The test results in Figure D2 and Table D2 show that the vast majority of interrupts are successfully responded in 2 microseconds. Remarkably, all interrupts are responded within 10 microseconds among all test cases. The average and maximum interrupt scheduling latency are 2 microseconds and 10 microseconds respectively. The results validate the effectiveness of the 6G RAN Cloud Platform for enabling the 6G RAN real-time processing.

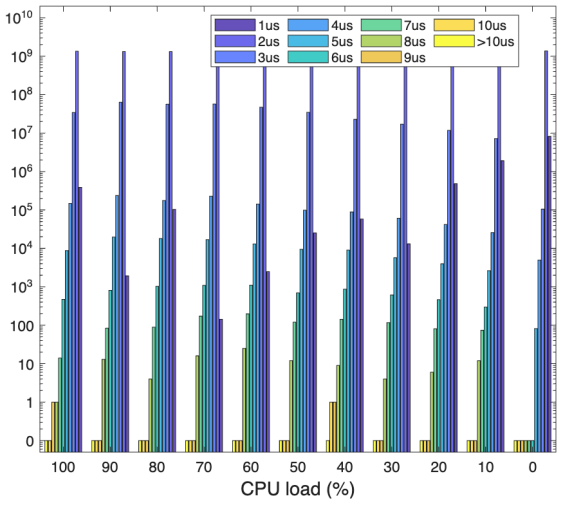

Figure D2 Scheduling latency distributions (logarithmic scale) under varying CPU load conditions (0% to 100%).

Table D2 Statistical properties of the interrupt scheduling latency distributions (shown in Figure D2)

Percentage	100%	90%	80%	70%	60%	50%	40%	30%	20%	10%	0%
Median (μs)	2	2	2	2	2	2	2	2	2	2	2
Mean (μs)	2.02	2.05	2.04	2.04	2.03	2.03	2.02	2.01	2.01	2.00	1.99
Maximum (μs)	9	8	8	8	8	8	10	8	8	8	5

This test validates that the platform's scheduling latency is within the Spatial Awareness requirement, sub-10 microsecond budget, a value derived from practical engineering experience to ensure sufficient time for RAN stack operations within a slot duration. Consequently, the test confirms the platform's capability to provide the deterministic real-time foundation necessary for Spatial Awareness, enabling distributed APs to function as a coherent, user-focused antenna array. Thus it effectively support UCN's seamless joint transmission.

Appendix D.2.2 *Clock synchronization test*

To fulfill the stringent synchronization requirement defined for Spatial Awareness in Appendix B—where coherent joint transmission and accurate environmental sensing in a distributed 6G UCN rely on multiple access points operating as a single synchronized entity—we conduct a clock synchronization test based on the Precision Time Protocol (PTP). This test evaluates the system's ability to achieve the essential sub-microsecond synchronization, particularly the $\leqslant 65$ ns time synchronization precision required to enable Spatial-aware transmission and sensing in UCN.

The test measured the PTP offset between two nodes in the 6G RAN Cloud Platform, each fitted with an Intel E810 PTP-capable NIC (enp44s0f1, PHC /dev/ptp1), and connected through the same L2 Ethernet fabric. The linuxptp-2.0 was installed on both nodes, where one node was configured as grandmaster by lowering priority 1 to 120, while the second node in slaveOnly mode. The ptp4l daemon was invoked with UDP/IPv4 transport, E2E delay mechanism and hardware time-stamping, logging at 1 s intervals. Synchronization success was defined as the slave port reaching the SLAVE state and maintaining an absolute master offset below the 1 µs budget specified for 6G fronthaul [24, 25]. It is shown in Figure D3.

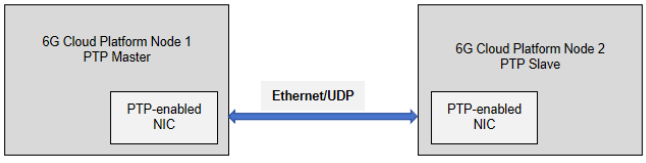

Figure D3 PTP test setup

As illustrated in Table D3, the clock synchronization process exhibits three distinct phases. In the initialization phase (0-1 seconds), a substantial offset of approximately 48 seconds was observed, which far exceeded the required synchronization precision. During the synchronization adjustment phase (2-10 seconds), the offset began to fluctuate and underwent rapid convergence—decreasing from 570 nanoseconds (at t=4 seconds) to -33 nanoseconds (at t=10 seconds), though not yet stabilized. The system then entered the synchronization steady phase (11-13 seconds), where the offset consistently remained within a narrow range (3 ns, 0 ns, 7 ns), well below 10 nanoseconds, thereby meeting the precise synchronization demands of the network.

Table D3 PTP offset during the synchronization process over time

Time (s)	0	1	2	3	4	5	6	7	8	9	10	11	12	13
Offset (ns)	48×10^9	48×10^9	-9	16	570	-117	-80	-145	-98	-103	-33	3	0	7

This sub-microsecond clock synchronization test confirms that the platform can meet the stringent synchronization targets (e.g., $\leqslant$ 65 ns time synchronization precision). This capability is fundamental to preventing inter-symbol and inter-carrier interference, thereby validating the platform's support for the coherent coordination required by UCN's Spatial Awareness.

Appendix D.2.3 *Container network performance test*

This test is designed to validate the high-throughput networking capabilities required by two critical aspects of UCN: the physical-layer fronthaul under Spatial Awareness, and the intensive intra-BBU data exchange for Context Awareness. It verifies whether the platform can support the massive, low-latency data flows essential for dynamic UCN topologies, such as those in cell-free massive MIMO systems where distributed units must collaboratively process user data and channel state information.

In this test, two widely adopted virtualization technologies, SR-IOV and OVS+DPDK [18,26], are evaluated for container networking. SR-IOV offers direct hardware access by bypassing the hypervisor, thus providing near-native performance, strong isolation, and minimal CPU overhead—making it suitable for latency-sensitive, high-throughput traffic. In contrast, OVS+DPDK performs packet processing in user space using software-based virtual switching. While it introduces higher CPU utilization and latency compared to SR-IOV, it offers greater flexibility, scalability, and support for complex network policies.

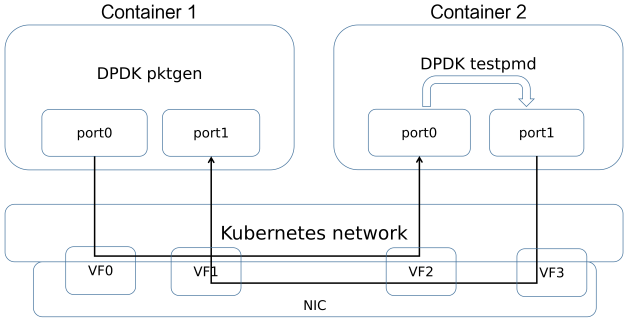

Figure D4 Network performance testing model for SR-IOV

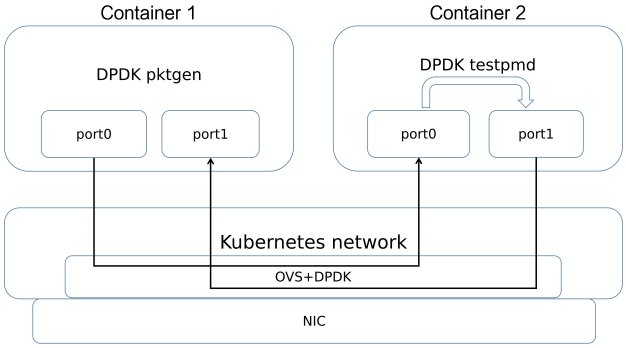

Figure D5 Network performance testing model for OVS+DPDK

The fundamental mechanism difference is revealed in their packet forwarding paths, as illustrated in Figure D4 and Figure D5. In the SR-IOV setup, packets travel directly between container Virtual Functions (VFs) on the NIC via PCIe, completely bypassing the host kernel and CPU. This creates a dedicated hardware path that minimizes latency. Conversely, in OVS+DPDK, all packets must traverse the OVS data path in the host's user space, requiring significant CPU cycles for packet parsing, matching, and forwarding decisions. This software-based approach inherently introduces higher latency but enables rich packet processing capabilities.

To quantitatively compare their performance, we conducted a dual-container test using DPDK's pktgen [27] as the traffic generator in container 1 and testpmd [26] as the forwarder in container 2. Each container was allocated 2 dedicated CPU cores to isolate processing resources. The evaluation covered the standard Ethernet frame size range from 64 to 1518 bytes defined by IEEE 802.3, including the header and trailer. We tested throughput (Mbits) and packet rate (Mpps) as the primary performance metrics. This comprehensive parameter space allows us to characterize performance across various traffic patterns.

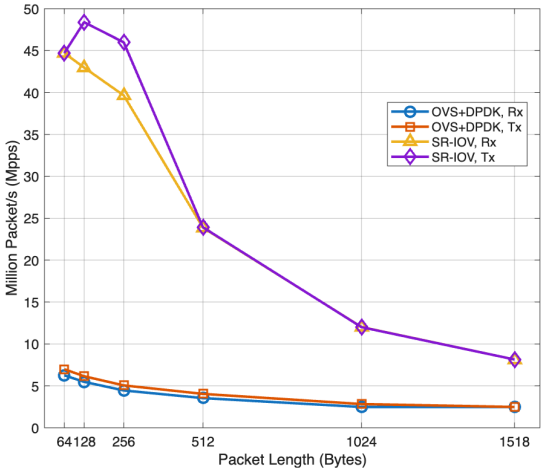

Figure D6 Packet rate (in Million packets per second, Mpps) across different packet lengths (64 to 1518 Bytes), comparing the SR-IOV and OVS+DPDK

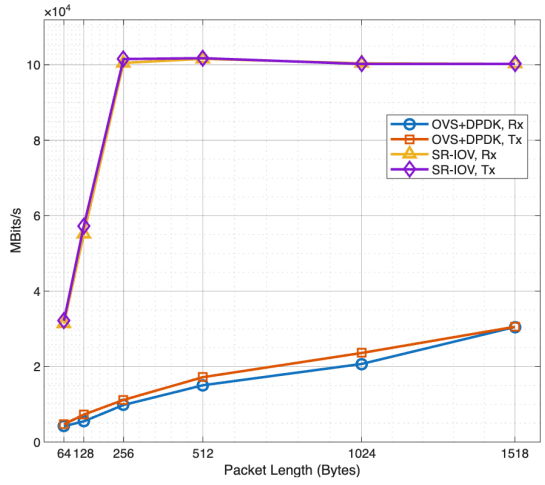

Figure D7 Throughput across different packet lengths (64 to 1518 Bytes), comparing the SR-IOV and OVS+DPDK

Figure D6 shows the results of pps tests with different packet lengths under two network testing models. The number of packets transmitted and received per second gradually decreases with increasing packet length because the total throughput of the network bandwidth reaches the maximum of 100 Gbps. Thus, the result of 1518 bytes decreases to around 10 million. In OVS-DPDK scenario, the number of packets transmitted and received per second is approximately 6 million pps at 64 Bytes packet length, and decreases to 2.5 million pps with packets length increases to 1518 Bytes.

Figure D7 shows the results of network throughput MBit/second tests with different packet lengths under two network testing models. The results show that SR-IOV can reach about 32 Gbps throughput for transmitted and received in 64 bytes scenario, and as the packet length increases to 256 bytes, the throughput reaches the line rate of 100 Gbps. In OVS-DPDK scenario, 64 bytes packet length can reach 4 Gbps, and as the packet length continues to increase, the throughput keeps growing to 30 Gbps for 1518 bytes packet length.

In this test, SR-IOV achieved a throughput of 100 Gbps, while OVS-DPDK only reached 30 Gbps. The main reason is that OVS-DPDK is heavily dependent on the CPU for packet forwarding. Thus it demonstrated SR-IOV is capable to support full 100 Gbps line rates, meeting both the physical-layer north-south fronthaul requirements for Spatial Awareness and the high-throughput east-west demands between BBUs for Context Awareness. The technology's hardware-assisted approach provides the deterministic performance needed for radio-rate traffic. However, OVS+DPDK, despite its lower raw performance, offers valuable flexibility through software-

based programmability. It remains viable for east-west traffic within the BBU pool where moderate throughput suffices and advanced network functions—such as service mesh, fine-grained traffic engineering, or multi-tenant isolation—are required. The platform's dual-plane networking architecture, incorporating both technologies, provides a balanced solution that addresses the diverse connectivity requirements of UCN workloads.

Appendix D.2.4 *AI capability validation and GPU power consumption test*

UCN requires computing resources, such as AI accelerators, to be dynamically managed around the user's context. This necessitates not only high performance, but also efficient power-driven elastic computing resource allocation and release. This test evaluates the platform's ability for dynamic and elastic resource management under Context Awareness, specifically for AI-native workloads.

The following GPU test is therefore more than an AI benchmark: it treats the graphics accelerator as an elastic, location-agnostic slice pool whose size and attachment point move with the user's service context. By visualizing physical GPU to MIG in the 6G RAN Cloud Platform, the semantic coding and decoding tasks (for data processing like training or inference) can shrink, expand or handover AI compute without ever breaking the perceptual "cell-free" experience. In short, the power curves below are the hardware footprints of UCN's promise: what travels is no longer just bandwidth, but silicon itself. Thus verifying the basic AI capability of the testbed and making it possible for the power consumption evaluation of the semantic communication system in the cloud environment.

The CPU, Random Access Memory (RAM) and container settings for the power consumption evaluation are listed as follows. For the GPU in the test, it is partitioned into 2 or 4 MIG instances, each MIG is fully isolated with its own high-bandwidth memory, cache, and compute cores.

- CPU: 5 physical core, Intel Xeon Gold 6348 @2.6GHz
- RAM: 64 GB
- Container network: two DPDK NIC
- GPU: NVIDIA A30 24 GB

The following test cases are further considered to assess the power consumption's of containers and GPUs. Each test case runs for one hour and the power consumption of each minute is recorded.

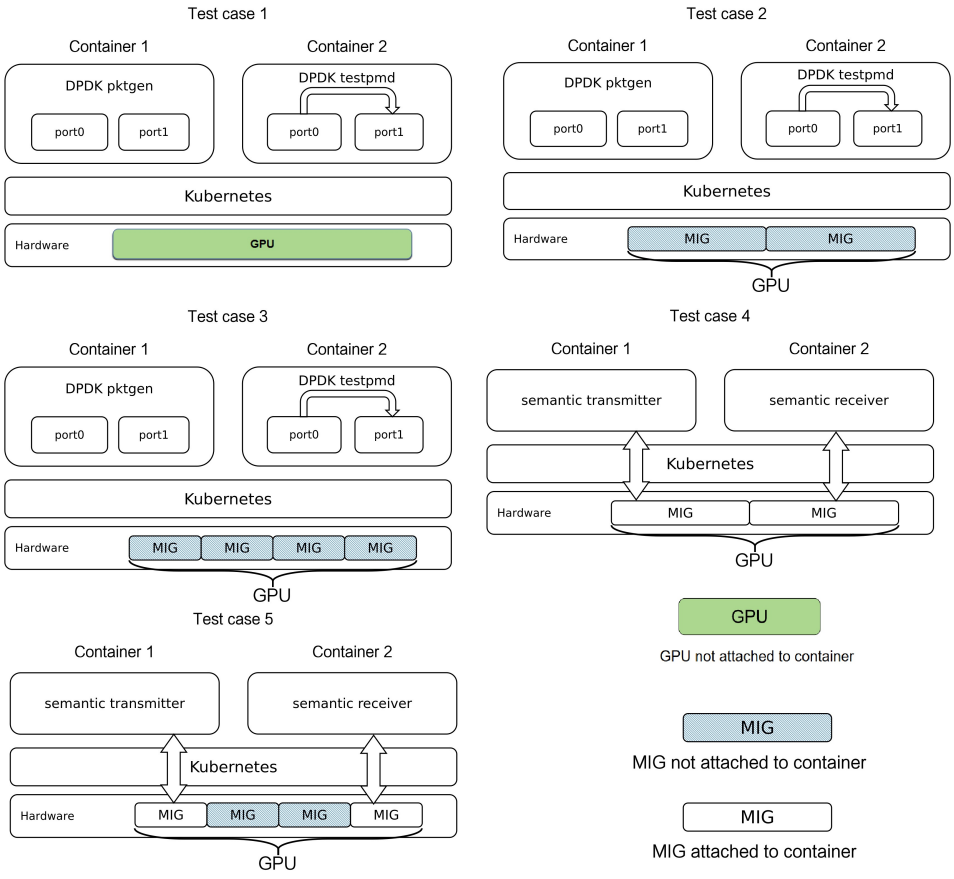

Figure D8 Power consumption test cases

- Test case 1: 2 containers running DPDK forwarding process, MIG disabled, GPU power on but not serving containers.
- Test case 2: 2 containers running DPDK forwarding process, MIG enabled with 2 MIG segmentation, 0 in used in total of 2.
- Test case 3: 2 containers running DPDK forwarding process, MIG enabled with 4 MIG segmentation, 0 in used in total of 4.
- Test case 4: semantic communication with MIG enabled with 2 MIG segmentation, 2 in used in total of 2.

- Test case 5: semantic communication with MIG enabled with 2 MIG segmentation, 2 in used in total of 4.

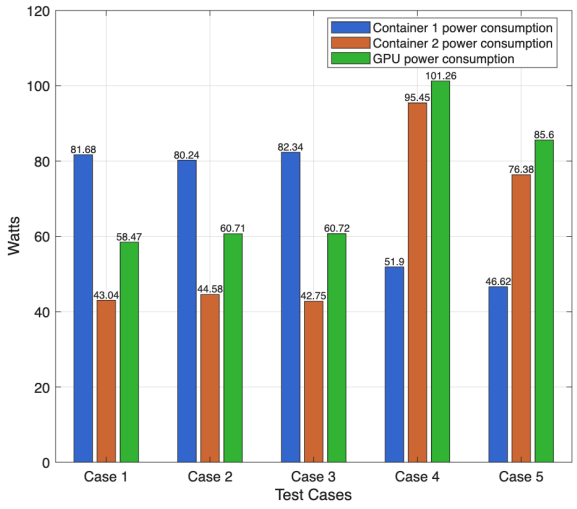

Figure D9 GPU power consumption test results

As illustrated in Figure D9, the average power consumption analysis reveals several key insights into GPU's behavior under different MIG configurations. Test cases 1 through 3, which involved containers running DPDK forwarding processes, demonstrate that simply enabling MIG or altering the number of partitions—without actually binding them to active containers—has no substantial impact on GPU power consumption. However, a clear GPU power increases from 60.71 to 101.26 Watts when MIG instances are actively bound to containers, as evidenced by the significant consumption jump in test case 4 compared to test case 2. More importantly, a comparative analysis between test 5 and test 4 reveals an approximately 15% reduction in GPU power consumption. This gain is attributable to a more efficient partitioning strategy: using four MIG instances and binding two of them to containers proves more power-efficient than creating and binding two larger partitions, despite the identical number of active containers. This indicates that matching the partition size precisely to workload requirements, rather than over-provisioning larger slices, contributes directly to reduced energy consumption.

The results collectively demonstrate that flexible MIG partitioning and on-demand resource binding are essential to minimize system power consumption when GPU acceleration is required by the services. By enabling fine-grained, right-sized allocation of visualized GPU resources, i.e., MIG, the platform directly supports the dynamic and elastic resource management mandated by Context Awareness in UCN. This capability not only helps improve energy efficiency, but also establishes a practical foundation for future research into intelligent power-driven computing resource scheduling for UCN.

Appendix E Future works

Looking ahead, future research will be pursued along two complementary directions. First, building upon the platform capabilities validated in this work, comprehensive end-to-end evaluations are beneficial to assess user-centric service performance, such as dynamic user association, seamless mobility, and intent-driven resource allocation—under realistic 6G deployment scenarios. Second, we will delve deeper into AI-native RAN through the tight integration of AI computing with RAN functions. This includes developing collaborative intelligence frameworks that span centralized and edge resources for real-time learning and control, enabling AI-driven radio resource management, closed-loop network optimization and Intent-aware autonomous orchestration. Furthermore, we will investigate multi-agent systems where distributed AI agents interact via standardized protocols (e.g., A2A and MCP) to interpret high-level user intents and autonomously orchestrate network actions across domains. These efforts will collectively establish the foundation for truly personalized, intelligent, and adaptive 6G connectivity.

Acknowledgements

This work was supported by National Science and Technology Major Project of China under Project GXB-2-2025-5.

References

1 Drampalou S F, Uzunidis D, Vetsos A, et al. A user-centric perspective of 6G networks: A survey. IEEE Access, 2024, 12: 190255–190294
2 Elhoushy S, Ibrahim M, Hamouda W. Cell-free massive MIMO: A survey. IEEE Commun. Surv. Tutor., 2021, 24: 492–523
3 Ammar H A, Adve R, Shahbazpanahi S, et al. User-centric cell-free massive mimo networks: A survey of opportunities, challenges and solutions. IEEE Commun. Surv. Tutor., 2021, 24: 611–652
4 Respati M A K, Lee B M. A survey on machine learning enhanced integrated sensing and communication systems: Architectures, algorithms, and applications. IEEE Access, 2024, 12: 170946–170964
5 Zhang J, et al. Phase and frequency synchronization for cell-free massive MIMO. IEEE Trans. on Wireless Commun., 2020, 19: 192–207
6 O-RAN Alliance. O-RAN fronthaul control, user and synchronization plane specification. Wg4.cus.0-v07.00, 2023

7 ITU-R M.2410: Minimum requirements related to technical performance for IMT-2020 radio interface(s), 2017
8 IEEE standard 1914.1-2019: Standard for packet-based fronthaul transport networks, 2019
9 eCPRI Specification. Common public radio interface: eCPRI interface specification. v2.0, 2019
10 3GPP. NR; Radio Resource Control (RRC); Protocol specification. TS 38.331, 3rd Generation Partnership Project (3GPP), July 2023
11 Björnson E, Sanguinetti L. Cell-free massive MIMO: A new paradigm for 5g and beyond. Foundations and Trends in Signal Processing, 2022, 15: 265–439
12 3GPP. Procedures for the 5G System (5GS). TS 23.502, 3rd Generation Partnership Project (3GPP), 2025-06. Release 19
13 Ksentini A, et al. Lightweight edge slice orchestration framework. In: Proceedings of Proc. IEEE Conf. NetSoft, 2023. 1–6
14 WG6 O R. Cloud platform reference designs, 2021
15 kernel.org. The Linux kernel archives, 2024. Accessed: 2024-05-18
16 Linux Foundation. Linux Foundation real-time linux wiki, 2024. Accessed: 2024-05-18
17 Open Container Initiative. Open Container Initiative, 2024. Accessed: 2024-05-18
18 Open vSwitch Project. Open vSwitch – production-quality multilayer virtual switch, 2024. Accessed: 2025-10-08
19 Cloud Native Computing Foundation. Kubernetes, 2024. Accessed: 2024-05-18
20 A. Celik, A.M. Eltawil. At the Dawn of Generative AI Era: A Tutorial-cum-Survey on New Frontiers in 6G Wireless Intelligence. IEEE Open Journal of the Communications Society, 2024, 5: 2433–2489
21 A2A Community. A2A Protocol — Agent-to-Agent communication, 2024. Accessed: 2025-08-18
22 Anthropic. MCP — Model Context Protocol, 2024. Accessed: 2025-08-18
23 Linux Kernel Contributors. rt-tests – real-time kernel test suite, 2024. Accessed: 2025-09-28
24 IEEE Std 1588^{TM} Series: Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems. IEEE SA Standards Board, 2025. Landing page for all versions and amendments
25 G.8275.1/Y.1369.1: Precision time protocol telecom profile for phase/time synchronization with full timing support from the network. Recommendation, Telecommunication Standardization Sector of ITU, July 2022
26 DPDK Project. DPDK – data plane development kit, 2024. Accessed: 2024-05-18
27 Pktgen Community. Pktgen-DPDK: High-performance traffic generator, 2024. Accessed: 2024-05-18