White Paper

5G Networking
Edge Computing

Streamlined Deployment of Private 5G Networks with Edge Services

The Private Wireless Experience Kit accelerates on-premises 5G implementations with reference blueprints based on Intel® Smart Edge Open, the edge computing software toolkit. Using a 5G core provided by the Hong Kong Applied Science and Technology Research Institute (ASTRI) and a 5G baseband unit (BBU) provided by SAGERAN, the Private Wireless Experience Kit offers a complete solution for private 5G and edge services on a single server. Customers can use this solution architecture in whole or in part as the basis for commercial offerings or their own deployments.

Authors

Intel:
Jiming Gao, Chuang Dong, Simon Li, Jayden Lee, Hongli Zeng, Feng, Yan, Xuekun, Hu, Hassnaa Moustafa, Mandar M Chincholkar

ASTRI:
Liang Dong, Jianjun Zhang, Yolanda Y.Y. TSANG, Xiaodong Wang, Zhicheng Luo, Xue Feng, Lawrence Luo

SAGERAN:
Ruixin Chen

Table of Contents

Making the best use possible of data is a key competitive differentiator for businesses of all types and sizes. An important part of meeting that challenge is the ability to process data wherever it delivers the most value, across the cloud, network, and edge. The high throughput and low latency of 5G networking are key enablers for emerging distributed computing models, especially for processing data at the edge, close to its source and the point of service delivery. The 5G-enabled edge enables real-time workloads and reduces bandwidth requirements for backhaul to the cloud or data center.

Cloud-native architecture provides workload portability across this distributed infrastructure, delivering network services using virtualized network functions (VNFs) based on microservices and deployed using containers. This approach enables network transformation with flexible deployment, rapid launch, and automated operation and maintenance.

The reference implementation introduced in this paper is a co-deployment of the 5G core, base station, and edge applications on a single server. Based on the Intel® Smart Edge Open Private Wireless Experience Kit and Intel Smart Edge Open, it uses the ASTRI 5G core and SAGERAN base station BBU deployed using network functions virtualization (NFV) infrastructure on Intel Xeon® Scalable processors. Deploying the 5G network and edge applications on a common hardware and software platform using a single physical host provides CapEx efficiency related to equipment procurement as well as OpEx efficiency for long-term maintenance of the solution.

1 Enabling On-Premises 5G with the Private Wireless Experience Kit

Enterprises streamline the integration and deployment of 5G functions and edge computing applications using the Intel Smart Edge Open Private Wireless Experience Kit. Enabled by the Intel Smart Edge Open toolkit, it delivers optimized performance for AI, video, and other services at the edge on Intel architecture. It supports low-cost, rapid deployment with a containerized solution. 5G data flows based on the Intel Smart Edge Open Private Wireless Experience Kit are illustrated in Figure 1.

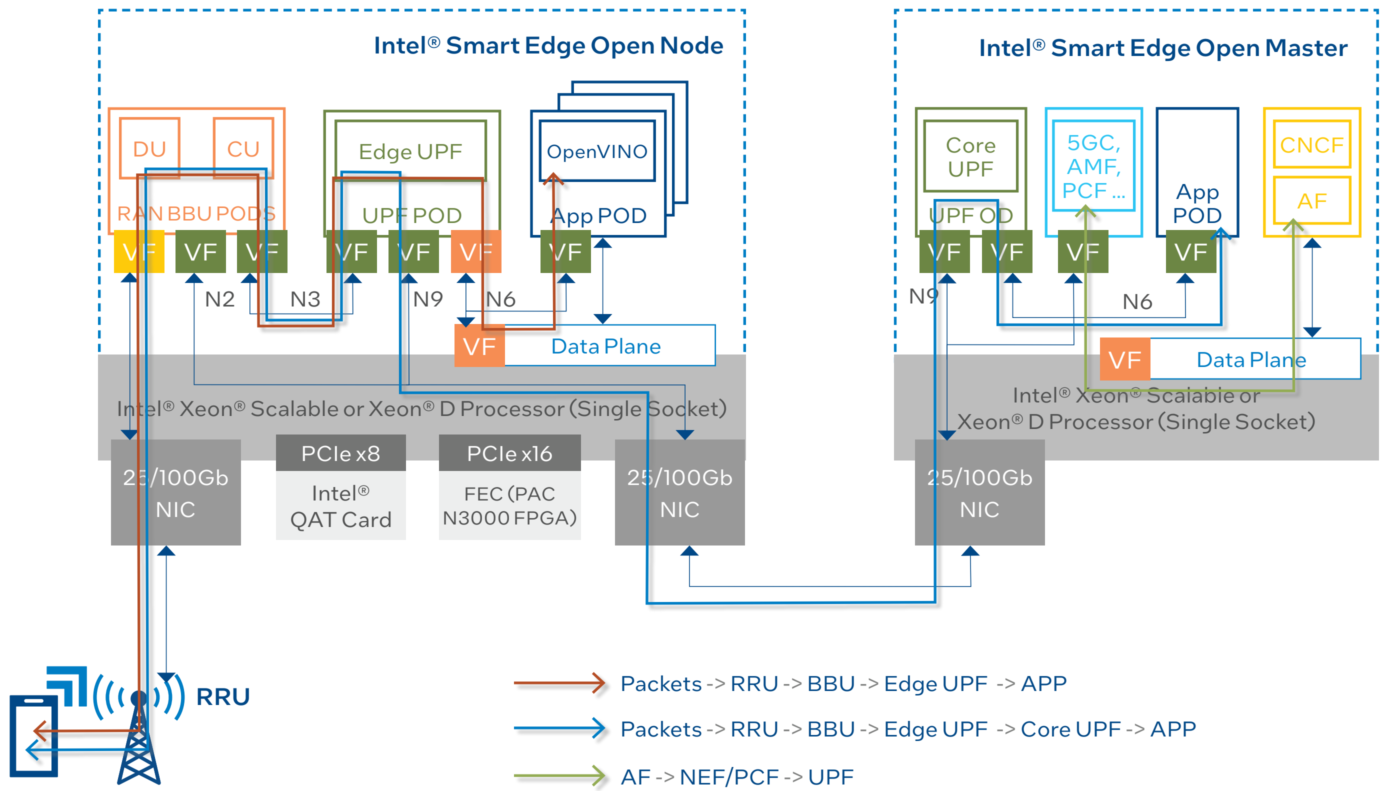

Figure 1. Data flows in the Private Wireless Experience Kit.

The Kit uses Intel Smart Edge Open to on-board and manage edge applications with high performance and agility. The reference implementation reported on in this paper also uses the ASTRI 5G Stand Alone (SA) Core and a base station solution based on SAGERAN 5G RAN. These components are described in more detail in the remainder of this section. Details of the hardware and software elements used in the reference implementation are summarized in Table 1.

Table 1. Hardware and software specifications of the Private Wireless Experience Kit reference implementation.

Hardware	**Server**	Intel® Server Board M50CYP Family (2U)
	Processor	2x Intel Xeon® Gold 6330N (20 cores, 2.20 GHz)
	Memory	128 GB DDR4-2933
	Network Interface Card (NIC)	Intel Ethernet Network Adapter XXV710-DA2
Software	**OS**	CentOS 7.9
	Kernel	RT Kernel 7.9
	Intel® Smart Edge Open	21.09
	Kubernetes	1.20.0
	Harbor	2.1.0
	Helm Chart	3.1.2
	Docker	20.10.2
	DPDK	19.11
	ASTRI 5G SA Core	21.06
	SAGERAN	21.03

1.1 Intel Smart Edge Open

Delivered as a set of royalty-free, modular building blocks, Intel Smart Open accelerates deployment of edge solutions, simplifies management, and increases quality and agility. The toolkit is built using Kubernetes technologies, with additional services to extend the control plane and edge node. It also provides adaptations and performance optimizations for Intel processors and acceleration technologies, with access to a broad and growing ecosystem of third-party partners and integrated solutions. The toolkit helps organizations scale edge solutions while maintaining platform consistency across the infrastructure.

Using Intel Smart Edge Open as a solution foundation provides a robust cloud-native architecture for the deployment of 5G VNFs and edge applications as containerized microservices. In particular, the reference implementation uses Intel Smart Edge Open capabilities to deliver the following innovations:

- **Orchestrate virtualized resources and services**, with support for virtual machines (VMs) and containers, as well as FPGAs, eASICs, and other intelligent acceleration cards, and performance optimization of north-south traffic and east-west traffic.
- **Deploy smart platform capabilities** based on Enhanced Platform Awareness (EPA), including configurations and capacity management across multiple orchestration stack layers. Specific functions include node feature discovery (NFD), CPU pinning and isolation in containerized scenarios, support for the discovery and allocation of large pages, and support for single-root I/O virtualization (SR-IOV).
- **Virtualize 5G base station BBU and 5G core network UPF**, with support for the deployment of VM and container solutions. Implement the corresponding solutions for the 5G network's multi-interface and fixed IP features.
- **Implement UPF support for the built-in DNS function**, which can realize DNS resolution and forwarding, making it easy to get through the DNS process of the 5G network and MEC.
- **Support self-developed AF function**, which can distribute MEC APP related shunt rules to the 5G network, with a 3rd Generation Partnership Project (3GPP) standards-compliant interface.

1.2 ASTRI 5G Core Network Solution

The ASTRI 5G SA Core is fully 3GPP compliant across interfaces, including HTTP2 service-based architecture (SBA) interfaces for network functions (NFs); examples are shown in Figure 2. Deployment options for the ASTRI 5G SA Core include support for the following capabilities:

- **Network slicing**: Sharing the same physical infrastructure for multiple unique logical and virtual networks with different topologies, supported features, and administrative domains.
- **Distributed UPF**: Supporting the N9 interface for intermediate UPF for offloading and steering traffic.
- **Local Area Data Network (LADN)**: Mobility control for specific locations for certain data networks (DNs).
- **Uplink Classifier (ULCL) and Branching Point (BP)**: Traffic control for different services and applications.
- **Session and Service Continuity (SSC) mode 1, mode 2, and mode 3**: Three different modes of IP anchor to provide continual support of applications with path updates.
- **Traffic steering and traffic redirection**: Control traffic at multiple levels based on configuring parameters for specific user equipment (UE).

The ASTRI 5G SA Core currently supports 3GPP Release 15, with Release 16 support continually evolving based on industry needs. All 5G core NFs are deployed and running as stateless microservices except the unified data repository (UDR), which provides a stateful context database for other NFs. Cloud-native 5G core network architecture optimizes the use of orchestration, deployment, and horizontal scalability enabled by Kubernetes.

1.3 SAGERAN 5G RAN Solution

The SAGERAN 5G base station is a mature, 3GPP-compliant commercial solution that can meet the requirements of mainstream operators. Its software architecture follows 3GPP requirements for centralized units (CUs) and distributed units (DUs), adopting an open and extensible compatibility design with standardized interfaces. Through simple configuration, the SAGERAN BBU can support either centralized or separated deployment of CU and DU, control plane (CU-CP), and data plane (CU-UP). It can also meet the specification requirements of different base station capacities for different business scenarios. The SAGERAN 5G microservice architecture is shown in Figure 3.

Figure 2. 3GPP compliance by the ASTRI 5G SA Core.

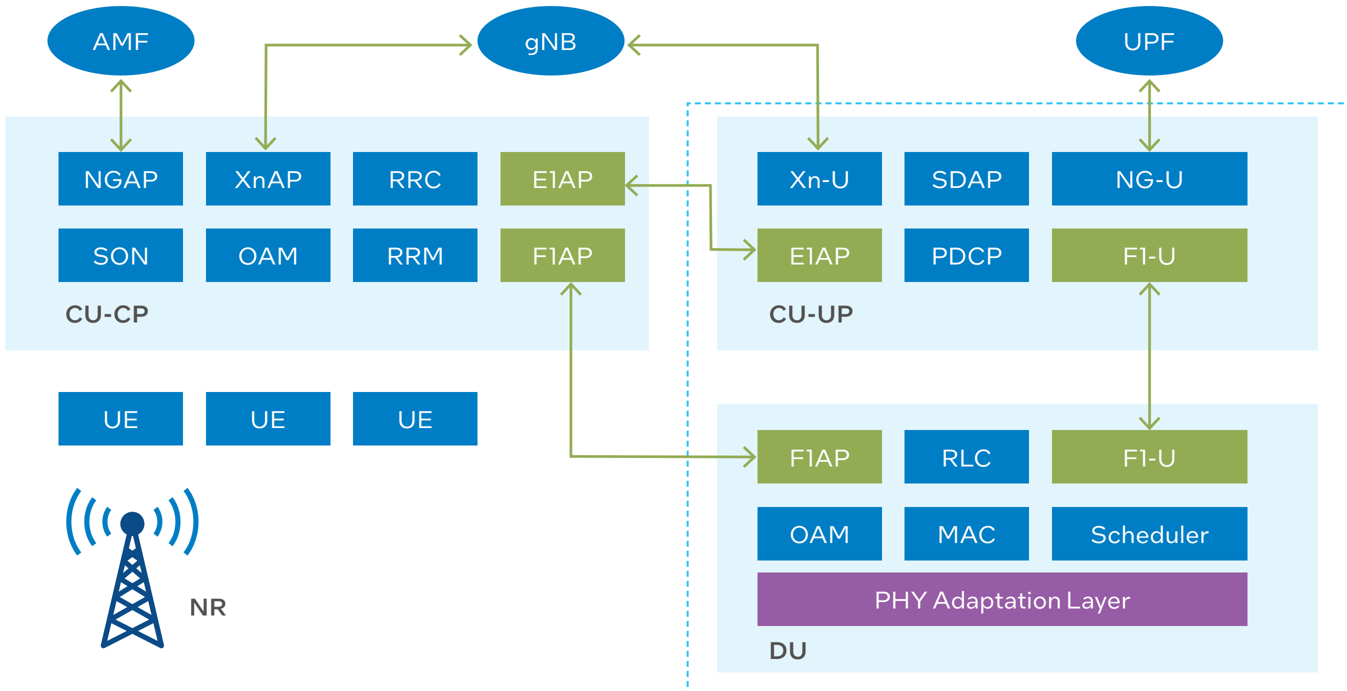

Figure 3. SAGERAN 5G microservice architecture.

2 Enabling Technology Building Blocks

The reference implementation draws on technology building blocks including those described in this section, which are supported by software components from the Intel Private Wireless Experience Kit (including Intel Smart Edge Open), the ASTRI 5G core network solution, and the SAGERAN base station solution. This full set of components is compliant with 3GPP standards.

2.1 NUMA Topology Manager

To ensure that throughput and latency requirements are met for sensitive applications, Kubernetes CPU Manager and Device Manager must be able to coordinate the resources allocated to individual workloads. Locality on specific non-uniform memory access (NUMA) nodes is of particular importance. For example, causing an instance of the 5G base station BBU software to utilize processor and network interface card (NIC) resources on separate NUMA nodes from each other can dramatically reduce throughput, introduce delays, and erode quality. The same is also true of other resources, such as accelerator cards.

NUMA Topology Manager enables NUMA alignment between the two processors in the server used in the reference implementation. Thus, for example, the solution can ensure that a workload container is placed on the same NUMA node where the NIC is provisioned and the memory is allocated. Topology Manager is a component of kubelet, which acts as a data source so that other components of kubelet can align resource allocation with NUMA topology.

2.2 Container Network Interface (CNI)

Standard Kubernetes exposes only a single, bridged interface to application or network pods, which creates limitations for network functions. Multiple network interfaces are typically required for usages such as separation of control, management, and data network planes, as well as to support different protocols or software stacks and different tuning and configuration requirements. The reference implementation enables the use of multiple network interfaces through CNI functionality added on top of the Kubernetes foundation, including Multus and Single-Root I/O Virtualization (SR IOV).

2.2.1 Multus CNI

Intel contributed the Multus CNI plugin to Kubernetes, which enables container pods to attach to multiple network interfaces, playing an important role in making NFV deployable in container environments. Integrated into Intel Smart Edge Open, Multus calls multiple other CNI plugins to enable those interfaces and provides a standardized method for specifying them.

2.2.2 SR-IOV CNI

SR-IOV can partition a single physical PCI Express device such as a NIC—referred to as a physical function (PF)—into multiple, independent virtualized devices called virtual functions (VFs) and then arbitrate resources among them. One or more VFs can be allocated to specific container pods for network or application workloads, and each VF has an independently configurable IP and MAC address. Resource management and packet exchange among VFs occur on the device hardware itself. SR-IOV offers the following benefits for Kubernetes:

- Direct communication with the NIC device achieves excellent performance close to "bare metal."
- Supports multiple concurrent fast network packet processing workloads in the user space, such as with the data plane development kit (DPDK).
- Uses NIC accelerator and workload offload.

The SR-IOV CNI plug-in enables Kubernetes pods to connect directly to the VF using the standard SR-IOV VF driver in the container host kernel.

2.3 Intel® Speed Select Technology

The reference implementation supports tuning the balance of performance and power efficiency on a per-workload basis. Using the native Kubernetes CPU Manager and NUMA Topology Manager, customers can specify CPU resources for a given pod. Intel Speed Step Technology (Intel SST) provides mechanisms to configure the hardware resources according to specific requirements.

Intel SST – Turbo Frequency (Intel SST-TF) controls settings to enable or disable Intel Turbo Boost Technology for CPU resources bound to compute-intensive pods, to increase compute throughput and improve performance. Intel SST – Core Power (Intel SST-CP) controls settings that govern CPU power state, so that less critical pods can be enabled to run at a lower power state, helping optimize power usage.

2.4 UPF Deployment Options

The reference implementation supports three UPF deployment options, as detailed in the remainder of this section.

2.4.1 Option 1: Passthrough + DPDK

- Both N3/N4/N9 and N6 run in Passthrough + DPDK mode.
- The physical NIC port is dedicated and has the best performance, but the connection between UPF and the MEC application requires an external connection.
- More physical NICs are needed to implement this solution.

2.4.2 Option 2: Bridge/SRIOV VF

- N3/N4/N9 works in Passthrough + DPDK mode, N6 works in bridge/SRIOV VF mode.
- Bridge/VF is used for the connection between UPF and MEC applications.
- This solution can reduce the number of physical NICs required.

2.4.3 Option 3: SRIOV VF + DPDK

- N3/N4/N9 work in SRIOV VF + DPDK mode.
- N6 also works in SRIOV VF + DPDK mode.
- This solution uses one physical NIC and has excellent forwarding performance.

2.5 5G RAN Deployment

The open, scalable software architecture used in the reference implementation provides flexible and convenient virtualization and container deployment. The implementation's 5G RAN microservice architecture is illustrated in Figure 4. Each unit of the base station, including CU-CP, CU-UP, and DU, can carry out virtualization and container deployment independently. Hardware resources such as CPU, FPGA, eASIC, and NIC capacity can be dynamically allocated using virtualization resources and services to improve utilization.

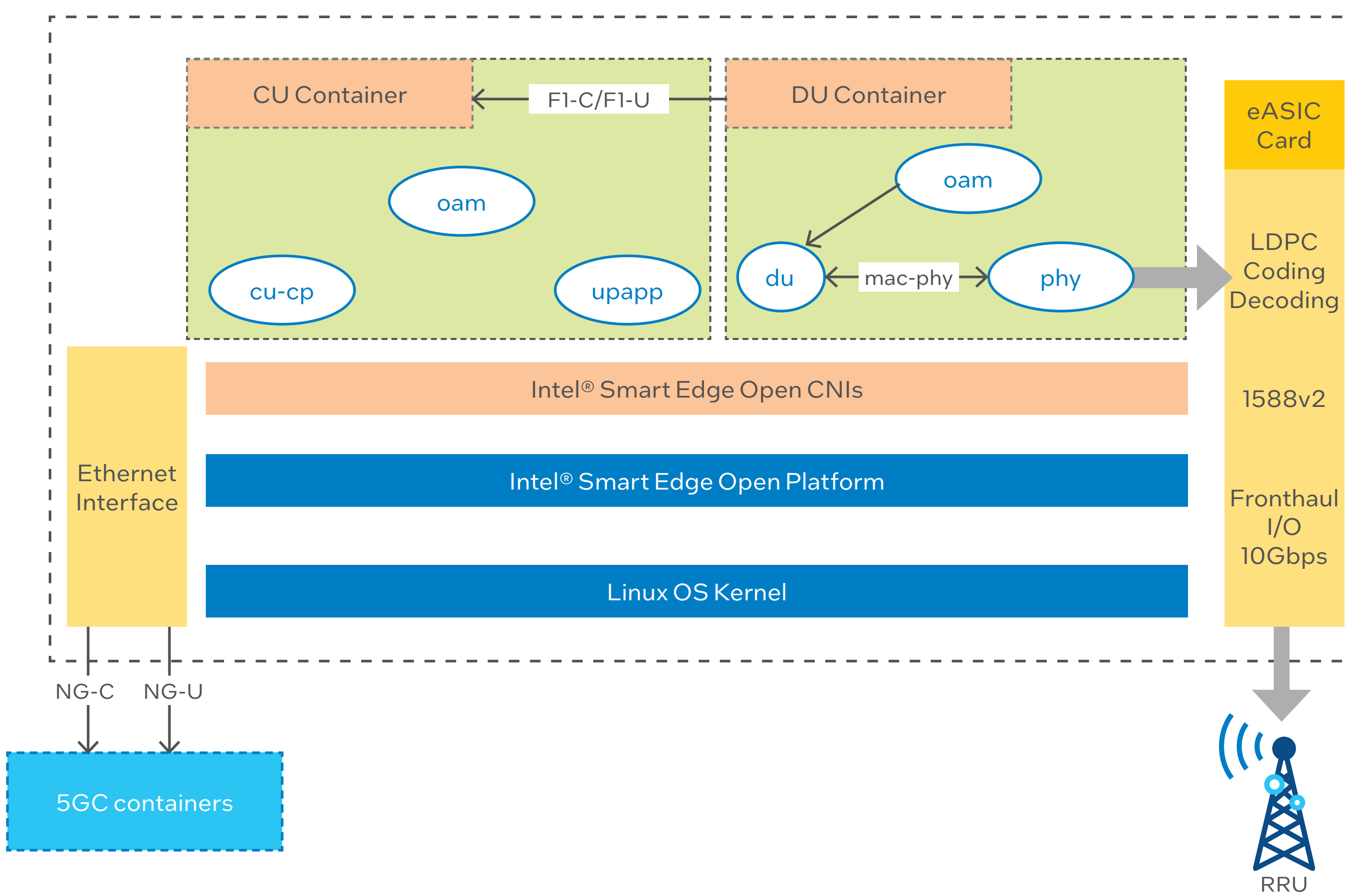

Figure 4. 5G RAN microservice architecture.

For scenarios that require centralized deployment of network elements, the ASTRI 5G SA Core can also adopt centralized common device deployment with the base station. Based on this "super integration" deployment scheme, it can provide complete 5G access services for end users under conditions of constrained hardware resources and power consumption. The reference implementation is designed to provide a total service throughput of not less than 800 Mbps downlink and 200 Mbps uplink. The display rate of the customer premises equipment (CPE) terminal is shown in Figure 5.[1]

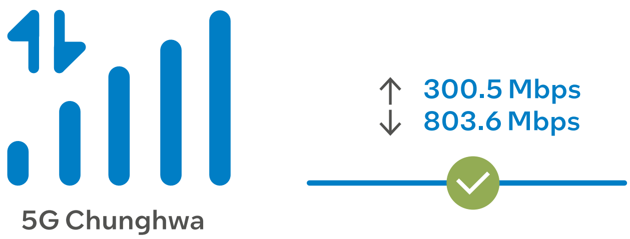

Figure 5. Customer premises equipment terminal display rate.[1]

3 ASTRI 5G Orchestrator

ASTRI 5G core network functions are deployed by the ASTRI 5G Orchestrator, represented in Figure 6, a cloud management platform based on the standard Kubernetes open-source API. The 5G Orchestrator is responsible for allocation of network and compute resources to virtualized 5G core network functions, as well as automated services such as configuration, deployment, performance monitoring, and fault reporting. A web management layer provides a graphical user interface (GUI) that enables operators to rapidly complete complex 5G network deployment tasks using simple drag-and-drop operations and instantaneously locate faults using network logs and powerful diagnostics.

In addition to supporting multi-location application deployment, the 5G orchestrator also supports flexible and fast deployment of 5G core network, edge UPF, and various edge IT/CT applications to enable innovation and the rapid launch of new services. The 5G Orchestrator also provides for elastic expansion of virtualized network functions to safeguard the performance of the core network, including during periods of peak service demand. Some key functions and features of the ASTRI 5G Orchestrator are described in the remainder of this section.

3.1 Automatic Configuration and Deployment of Network Functions

Lifecycle management tasks such as NF deployment or termination can be performed using the 5G Orchestrator web UI, represented in Figure 7. NF configurations can also be dynamically customized using the web interface to provision 5G NFs on demand, in real time. Right-clicking one of the green icons shown in the figure, for example, provides a visualization of the corresponding NF, while making connections among various 5G interfaces enables operators to perform flexible networking deployments.

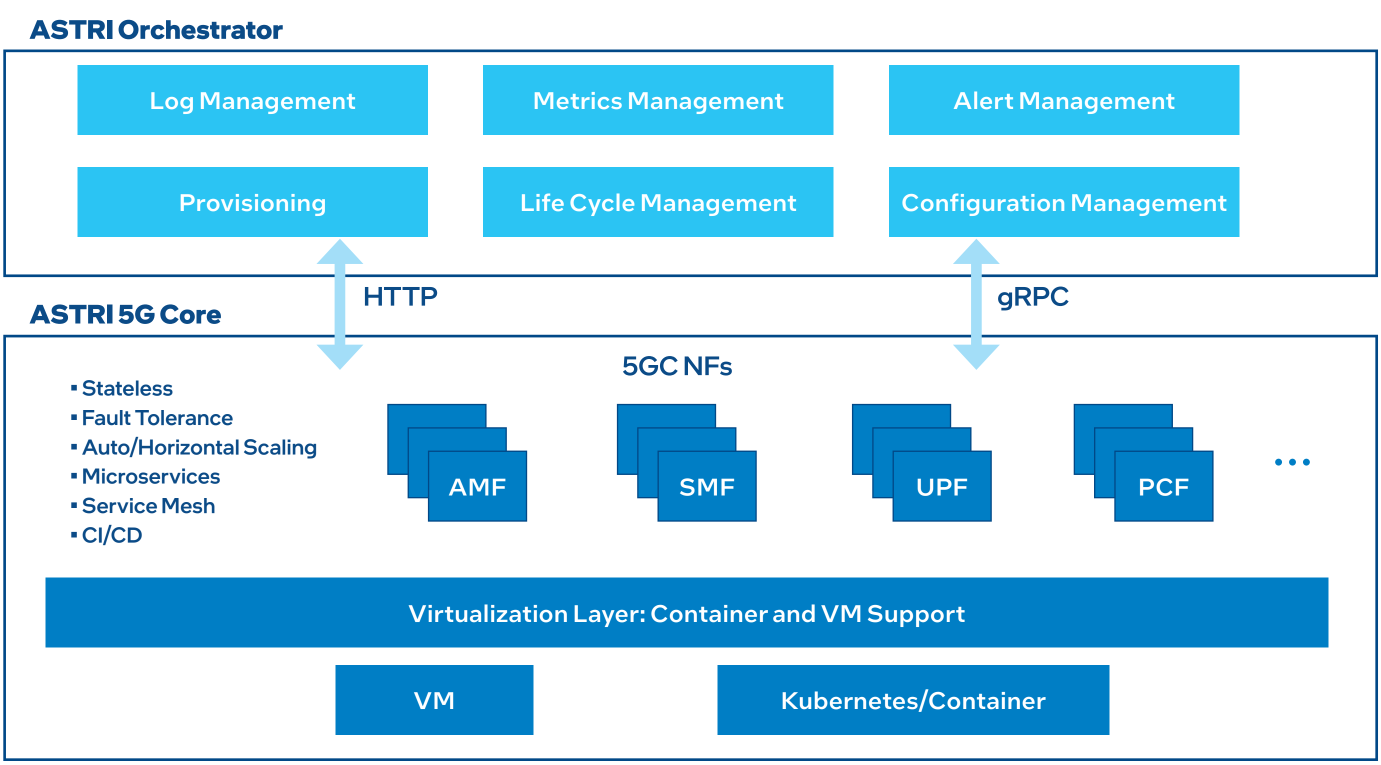

Figure 6. ASTRI 5G orchestrator.

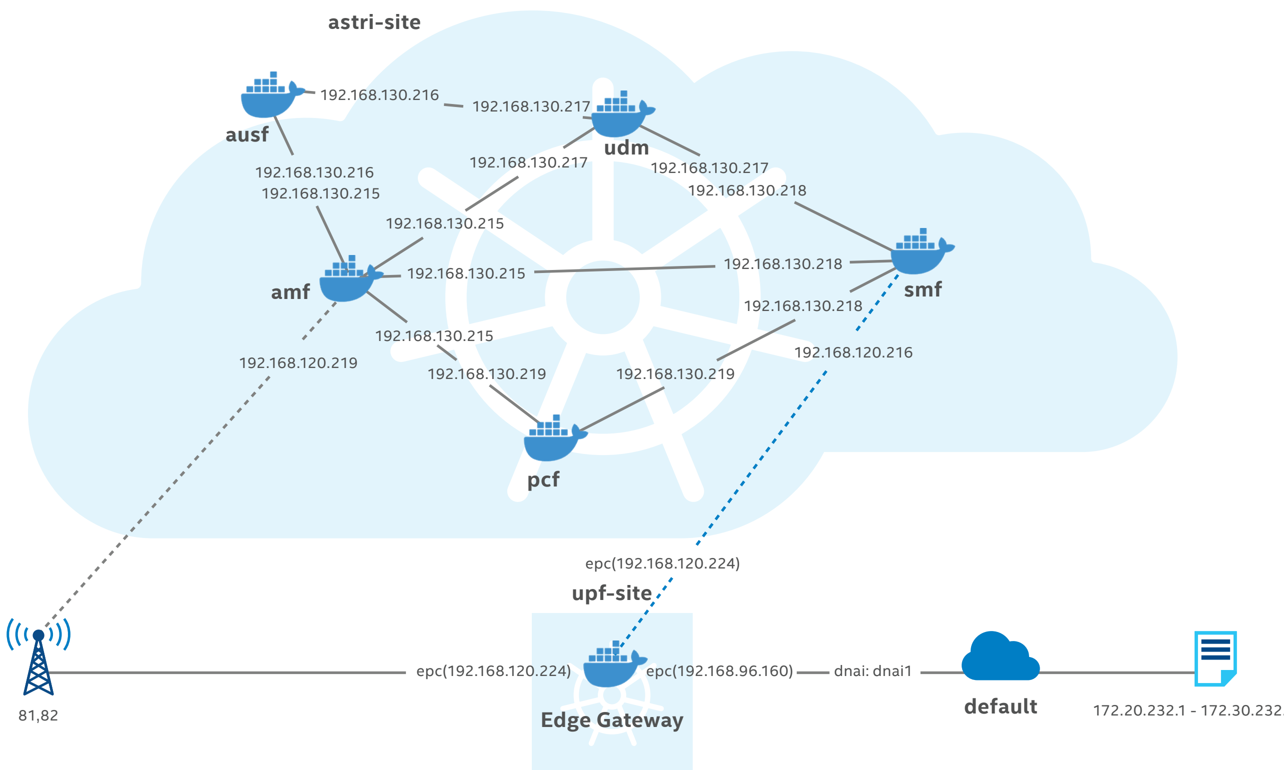

Figure 7. ASTRI automatic configuration and deployment of network functions.

3.2 5G Network Slicing Function

By defining independent logical networks that are each tailored to specific application service level requirements, 5G network slicing is critical to providing flexible, scalable network services. 5G Orchestrator provides agile slice management, automatically generating network slice configurations based on operator inputs. It also defines VNFs to provide required characteristics such as data isolation and provides northbound interfaces that enable end-to-end control using the web management interface.

3.3 Real-Time Data Monitoring Function

The 5G Orchestrator real-time data monitoring function collects and visualizes key performance indicators (KPIs) on NFs and users. The UI provides dashboards that represent the data using elements such as gauges and numeric metrics, as shown in Figure 8. As shown in the figure, 5G Orchestrator can monitor current CPU and memory usage or network throughputs associated with individual NFs. It also reports on the numbers of gNodeB, PDU, and UE sessions after successful 5G end-to-end session establishment.

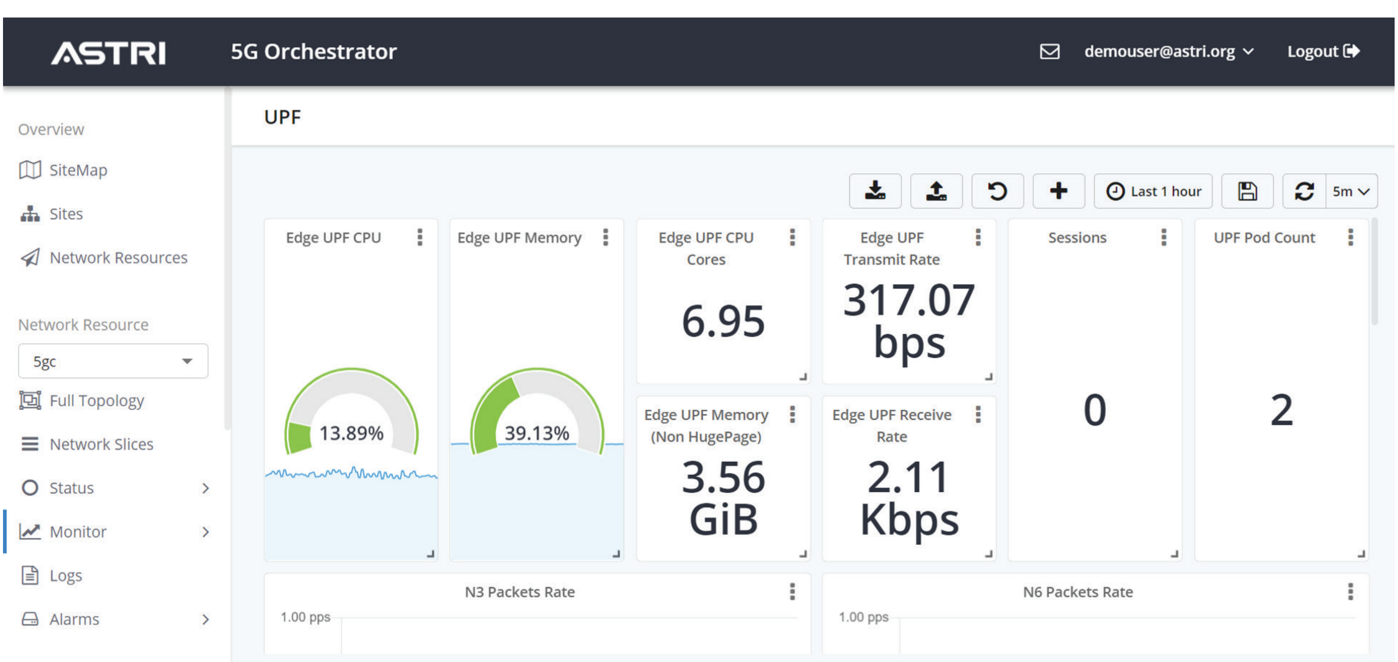

Figure 8. UPF real-time data monitoring.

3.4 Network Function Alerts

The 5G Orchestrator NF Alert function collects and displays alarm information related to 5G network elements in real time, as shown in Figure 9.

There are three types of NF alerts, triggered in the following circumstances:

- When interfaces between NFs are disconnected (for example, disconnection at the N1 interface between the AMF and base station).
- When heartbeats between NFs and the web management system are terminated.
- When real-time KPIs associated with an NF exceed specified thresholds.

4 Conclusion

The reference implementation described here shows the viability of deploying 5G NFs and edge applications on a single physical host, in compliance with 3GPP standards. It provides a design pattern that customers can use as a whole or in part as they develop commercial offerings for forward-looking edge computing infrastructure. By unifying and validating the solution stack based on the Intel Smart Edge Open Private Wireless Experience Kit, Intel Smart Edge Open, ASTRI 5G SA Core, and SAGERAN 5G base station on Intel architecture, this reference implementation enables customers to implement new design patterns quickly, accelerating solution time to market while increasing quality.

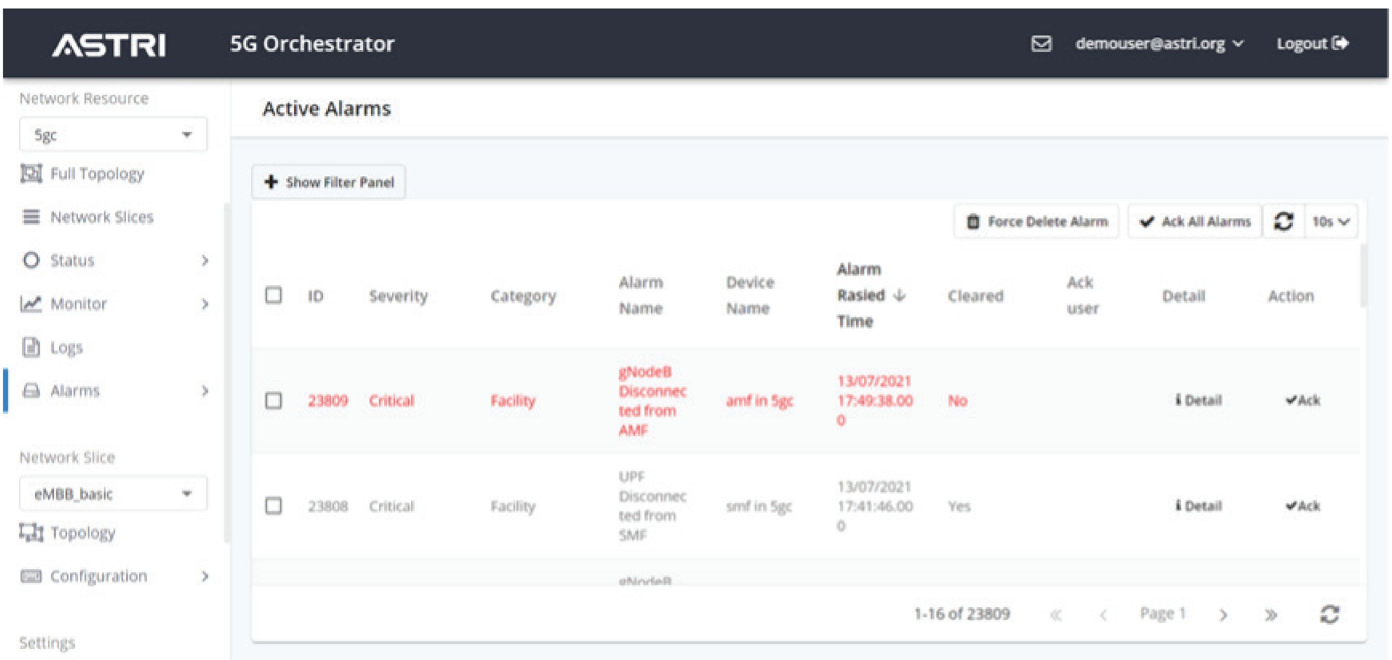

Figure 9. Active Alarms screen in 5G Orchestrator.

More Information

Hong Kong Applied Science and Technology Research Institute (ASTRI) – astri.org

Intel Network Builders Solution Brief: "5G User Plane Function (UPF) Performance with ASTRI" – networkbuilders.intel.com/solutionslibrary/5g-user-plane-function-upf-performance-with-astri-solution-brief

Intel Ethernet – intel.com/ethernet

Intel Smart Edge Open Overview – intel.com/content/www/us/en/developer/tools/smart-edge-open/overview.html

Intel Smart Edge Open GitHub – smart-edge-open.github.io

Intel Xeon Processors – intel.com/xeon

Private Wireless Experience Kit – smart-edge-open.github.io/docs/experience-kits/private-wireless-experience-kit

SAGERAN – sageran.com

[1] Attribution needed using the following format – Configuration: Baseline: 1-node, 2x <cpu> on <platform> with xxx GB (16 slots/ 32GB/ 3200[run at 2933]) total DDR4 memory, microcode 0x280, HT on, Turbo on, <OS>, <kernel>, 1x <S3700 400GB SSD>, <benchmark/workload version>, <compiler if used>, <library if used>, <other sw dependency if used>, test by <company> on <mm/dd/yyyy>. New: 1-node, 2x <cpu> on <platform> with xxx GB (16 slots/ 32GB/ 3200[run at 2933]) total DDR4 memory, microcode 0x280, HT on, Turbo on, <OS>, <kernel>, 1x <S3700 400GB SSD>, <benchmark/workload version>, <compiler if used>, <library if used>, <other sw dependency if used>, test by <company> on <mm/dd/yyyy>.

0422/FZ/MESH/346465-001US