Microsoft programmable RAN platform with dynamic service models

Contents

Executive summary

Virtualization of telco networks is beginning to happen in Radio Access Networks (RAN), especially regarding Open Radio Access Networks (O-RAN). However, there are some obstacles to this trend. The current O-RAN architecture limits programmability to predefined telemetry and control that an xApp can access through existing, standardized service models which may limit the pace of innovation.

To address this limitation and unleash O-RAN's full potential, Microsoft has developed programmable RAN platform technology that introduces flexible, dynamically loadable service models. The proposed architecture is primarily based on the existing O-RAN architecture. The main difference is the dynamic service model, whose functionality can be implemented by the application designer and deployed at run-time without affecting RAN operations. All this flexibility does not come at the cost of reliability, safety, or security.

New applications of analytics and automation are possible with the flexibility of dynamic service models and low latency, including RAN energy efficiency and anomaly detection. New applications are able to access almost all information available at different layers of RAN and exercise control at many different levels of RAN. As a result, Open RAN has the potential to significantly accelerate the pace of RAN transformation, making it possible to achieve the full benefits of 5G sooner.

"Open RAN has the potential to significantly accelerate the pace of RAN transformation, making it possible to achieve the full benefits of 5G sooner."

Introduction

The mobile cellular network continues to evolve as the world moves from 4G to 5G and beyond. The drive for more secure, reliable services from consumers exploded with 4G. 5G further addresses use cases requiring higher throughput and lower latency with improved network efficiency and device density. In addition to operators' macro networks serving consumers, 5G has unlocked the potential of industry 4.0 use cases that require reliable and low latency network connectivity. 5G architecture is inherently based on microservices and is cloud-native, which paves the way to leverage the benefits of virtualization.

Virtualization of telco networks started with packet core and is beginning to happen in Radio Access Networks (RAN). A RAN is a network that connects mobile devices to the core network of a cellular service provider. RAN is an essential part of any cellular network, and it consists of a series of base stations connected to user phones via radio waves. RAN plays a critical role in providing coverage and capacity for a cellular network. It is often designed to be scalable so that it can be easily expanded as the needs of a network change.

5G furthers the path to virtualize this RAN network that can be deployed on commercial off-the-shelf servers. O-RAN furthers this approach with open interfaces between RAN components and introduces new components to enable programmability. Open RAN, through the O-RAN Alliance, defines the architecture and introduces new interfaces and network elements. The new network elements are Near-Real Time RAN Intelligent Controller (nRT-RIC), Service Management and Orchestration (SMO), and Non-Real Time RIC. The new network interfaces are O1, E1, A2, O2, and Open Fronthaul (OFH).

O-RAN promises disaggregation of functions, enabling full programmability and new services while meeting latency and network automation requirements that were not achievable in the past. This can be achieved if there is the ability to get telemetric data out of the different RAN layers and use it to build Apps for overall network monitoring and performance optimization.

This white paper demonstrates how an enhanced framework, including RAN telemetry, analytics, and artificial intelligence (AI) can further unleash the potential of programmable network operations, performance, and efficiency. A fully virtualized RAN software implementation that allows intelligence to be extracted at all levels––including layer 1 (L1)––is a key enabler of enhanced RAN Telemetry.

O-RAN overview

Traditional RAN solutions based on purpose-built hardware were designed with predefined design goals with compute capabilities that could not be augmented without replacing the installed solution. Virtualized RAN (vRAN), on the other hand, is composed of commercial off-the-shelf (COTS) hardware allowing the infrastructure to easily be modified and improved and making the opportunity for extensible, software-based RAN a reality.

Intel provides a vRAN reference architecture of the L1 of the RAN stack in the form of the FlexRAN™ reference software stack. Intel's FlexRAN™ reference software demonstrates how to optimize virtualized distribution unit (VDU) software implementations using Intel® C++ class libraries, leveraging the Intel® Advanced Vector Extensions 512 (Intel® AVX-512) instruction set.

The multi-threaded design allows a single vRAN software implementation to scale to meet the requirements of multiple deployment scenarios, scaling from single small cells deployments, optimized Distributed-RAN (D-RAN) deployments, or servicing a large number of 5G cells in Centralized-RAN (C-RAN) pooled deployments. As an software implementation, it can also support LTE, 5G narrow band, and 5G massive multiple input/multiple output (MIMO) deployments, all from the same software stack using the O-RAN 7.2x split. The FlexRAN™ reference software solution framework by Intel is shown in *Figure 1* below.

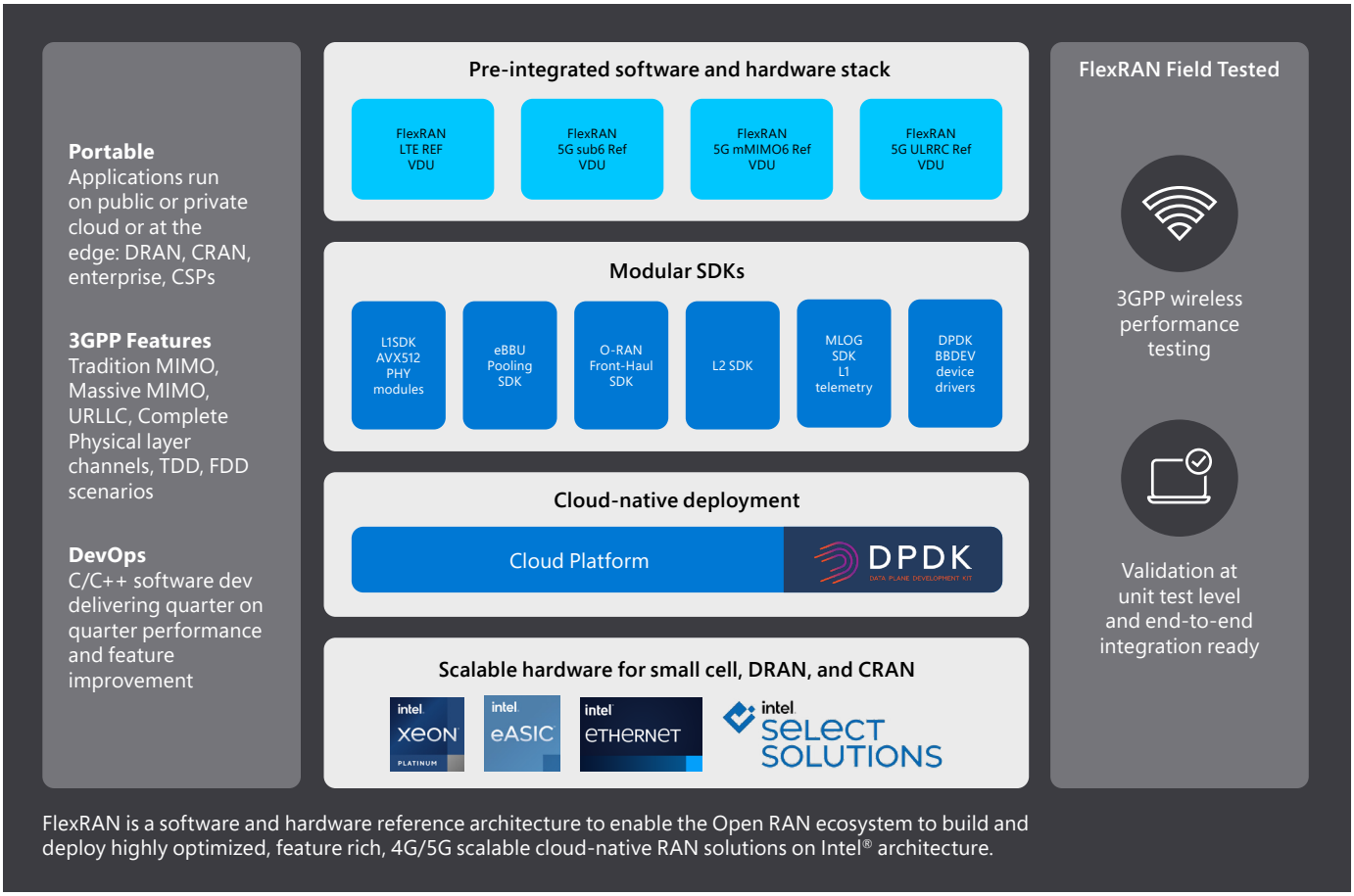

Figure 1: FlexRAN™ technology overview

Capgemini's software stack of Layer 2 (L2) and Layer 3 (L3) works seamlessly with Intel's FlexRAN™ reference software as Containerized Network Functions (CNFs). Integrating independent RAN layers from best-in-class providers is only possible with vRAN. As all RAN layers, including the central unit control plane (CU-CP), central unit user plane (CU-UP), distributed unit (DU), and the physical layer (PHY), can be deployed as software in a container, unlocking the ability to capture, correlate and analyze data in a much more granular and programmatic manner.

O-RAN offers an opportunity for new metrics and analytics with near real-time RIC (nRT-RIC) and the corresponding E2 interfaces and their service model. In this architecture, developers can build custom xApps that can access a rich telemetry stream from a software-based RAN and exert control over its behavior in a fine-grain manner. These capabilities allow for new service models that were not possible with the legacy, hardware-centric RAN architecture.

Non-real-time RIC further enables developers to leverage non-RAN data to provide intelligent network solutions. This data can be weather or sports event information or another data type. For example, a weather information-based solution could automatically adjust network settings based on the current weather conditions. This capability would ensure the network is optimally configured for the current conditions and would provide a better user experience. However, there are some roadblocks to this reality.

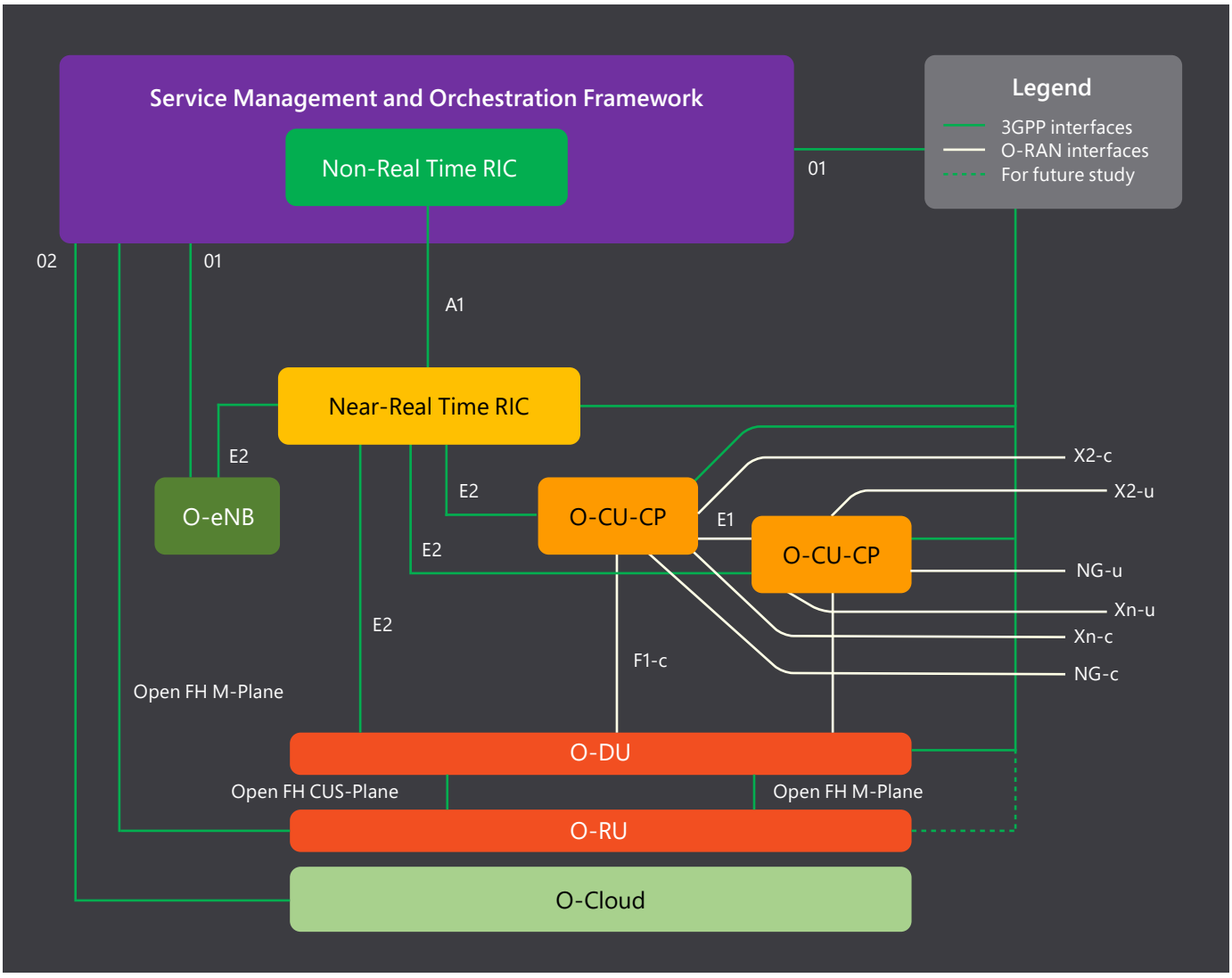

Figure 2: O-RAN high-level architecture

Limitations of current O-RAN analytics

The programmability of the current O-RAN architecture is still limited to predefined telemetry and control that xApps access through existing, standardized service models. The application programming interfaces (APIs) are called E2 service models and are standardized as a part of O-RAN. An xApp can access data through (a few) existing, standardized service models. An xApp vendor conceiving a new application without an appropriate service model must first attempt to introduce one through a long O-RAN standardization process and then wait on RAN vendors to implement support for the model as RAN functions in CU/DU. This can be a slow process and limits the rate of innovation.

The ability to get telemetric data out of the different RAN layers in a dynamic fashion goes beyond what is specified in O-RAN today. Also, O-RAN RIC architecture only supports near-real-time use cases with latencies of 10 milliseconds (ms) and above, which prevents app designers from innovating in many vital areas that require low latency, such as improving platform power efficiency, reducing scheduling latency, and improving spectral efficiency through fast channel prediction.

Take an example of a developer that has developed a novel channel estimation model that can significantly improve the capacity of the RAN. To implement this model as an xApp, the developer has to get access to raw channel data from L1, process them using their novel algorithm, and feed them back into the L1 decoding process. This poses several challenges.

First, the channel information at L1 has a very high volume of data, and it may be infeasible to collect all of it. The developer may wish to define a new way to aggregate this data or to send updates only when the channel sufficiently changes. But this is possible only if the logic to transport the data and define the control actions are added to a standard O-RAN service model. Besides waiting for the standardization process, the developer may need to relinquish a part of their proprietary code to the standard, potentially hampering their business model.

Second, the developer may want to collect information from several parts of L 1, including synchronization channels, sounding, and other reference signals. The developer needs to specify a different service model or schema for each information source in this case. And suppose another innovative developer wants to implement a different algorithm in a slightly different way. In that case, they may need to define a new set of service models, increasing the standardization and implementation complexity.

Similar data collection and control problems persist across the RAN L2 software (or L3 software). Detailed RAN telemetry may give access and control over various queueing delays, scheduling algorithms, power, timing adjustments, and much more. Today, it is difficult for developers to access this information and implement their products.

These limitations are addressed by the use of Dynamic service models which extends this Standard-yet-static interface. It introduces the capability of getting detailed internal states and real-time telemetric data out of the live RAN software in a dynamic fashion for new RAN control applications. With this technology, together with detailed platform telemetry, operators can achieve better network monitoring and performance optimization for their 5G networks, and enable new AI, analytics, automation capabilities that were not possible before.

Microsoft's programmable RAN platform framework

To address these issues and unleash the full power of innovation in the O-RAN, Microsoft has developed a RAN and platform analytics and automation technology with dynamic and flexible service models. We worked closely with Intel and Capgemini, whose virtualized and O-RAN solutions are implemented entirely in software, making them well-placed to introduce more openness.

Microsoft RAN and platform analytics speeds innovation by introducing flexible, dynamically loadable service models. We develop a system in which each developer can––in a safe and controlled way––upload their own or 3rd party custom code implementing a service model directly onto a live RAN platform.

A virtualized RAN process is a time-sensitive application, and to do any modifications we need to ensure that it does not cause any performance deterioration. We leverage existing open source and internally developed technologies to help ensure safety in this approach.

Dynamic service models

Dynamic service models will further enhance the capability of O-RAN RIC and help to accelerate the development of new use cases and xApps from the O-RAN solution perspective. At Microsoft, we will work with the O-RAN community to adopt them.

Advanced service models may require access to high-volume data streams, such as digitalized radio waveform samples, radio link control/medium access control (RLC/MAC) queue sizes, and packet retransmission information. These streams must be aggregated and compressed before being delivered to a xApp. Yet, xApps may require different aggregation algorithms, which can be supported dynamically in the proposed solution.

Moreover, a similar approach can be used to add real-time control algorithms from a marketplace. Dynamic service models allow developers to easily generate new real-time telemetry which can be used by a control application running at the far-edge in a safe environment to make control prediction and decisions at millisecond granularity. This further enables a class of important new applications currently not supported by O-RAN RIC architecture, which in the future may make a case for another type of RIC, a real-time RIC supporting use cases less than 10ms latency.

The collected data can be transferred to an application implementing a novel service or optimization running at a different location. This can be a xApp, receiving data over the E2 interface.[1] It can also be transmitted directly to a cloud for more advanced and powerful AI/ML analysis and processing. It can also be processed locally, at the far-edge, to achieve a high-speed, real-time control loop. As we illustrate through use cases later, this allows the deployment of machine learning (ML) models in a far-edge that can be used in scheduler algorithms, physical resource blocks (PRB)/slice allocation handling, and channel estimation algorithms with ML models co-located in real-time RIC.

1. Once the specification defines the ability to ship the "collected" data.

Finally, these dynamic service models also allow controllers to execute one of the carefully curated sets of actions on the RAN to change its behavior. With this, we close the full automation loop that adds flexibility at all levels of the networking stack, opening up the full software-defined potential of Open RAN.

The proposed architecture is illustrated in *Figure 3* below. It is largely based on the existing O-RAN architecture. The main difference is the dynamic service model (D), whose functionality can be implemented by the application designer and dynamically deployed at run-time without affecting RAN operations. This enables operators and trusted third-party developers to write their own telemetry, control, and inference pieces of code (called "codelets") that can be deployed at runtime at various points in the RAN software stack, without disrupting the RAN operations. The codelets are executed inline in the live RAN system and on its critical paths, allowing them to get direct access to all important internal raw RAN data structures, to collect statistics, and to make real-time inference and control decisions.

We also added a fast, real-time controller to support new applications enabled by the dynamic service model, requiring a sub-10 ms control loop. Some example applications are listed in the next section.

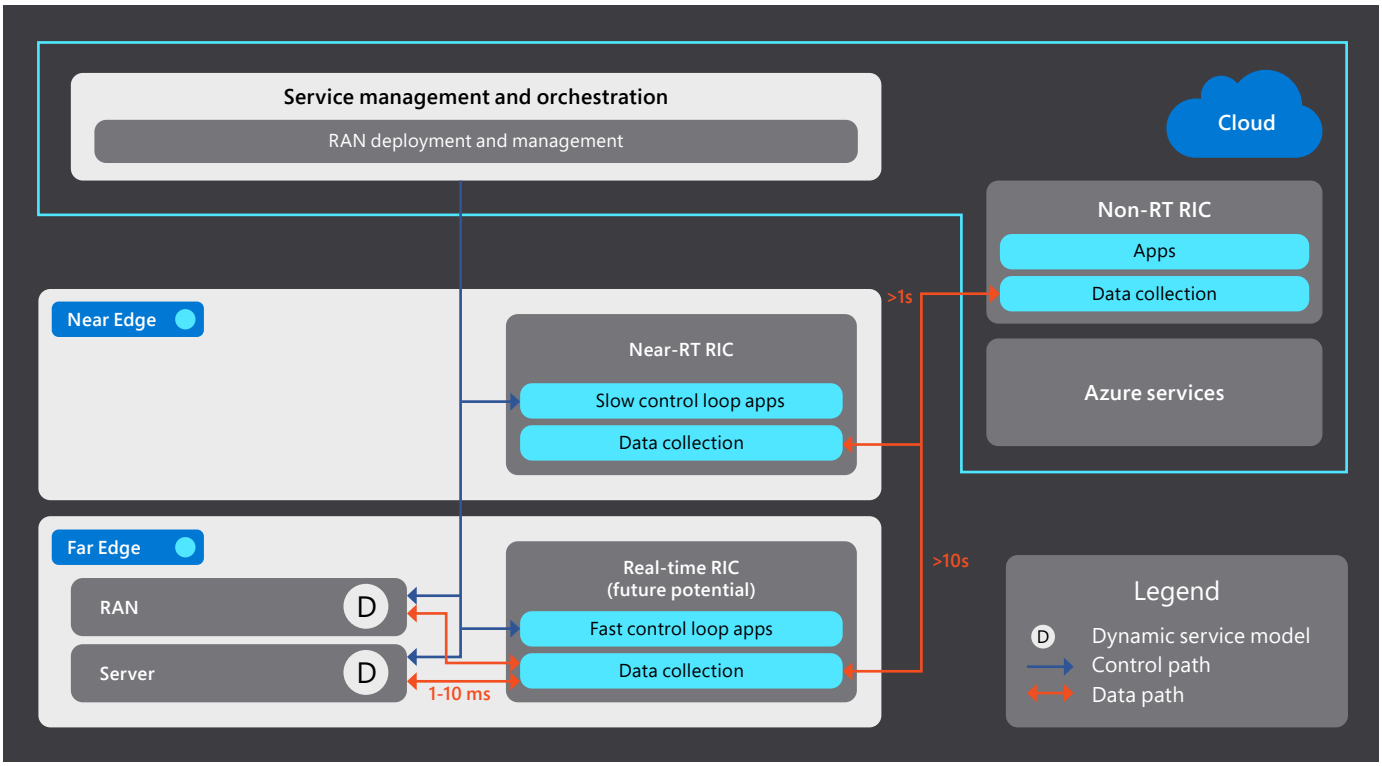

Figure 3: Proposed programmable RAN platform architecture

It is important to note, that all this flexibility does not come at the cost of reliability, safety, or security. The analytics and automation framework is carefully designed with these criteria in mind. The code used to define dynamic service models is statically verified using the latest state-of-art verification technologies. It relies on the same tools used to run user space Extended Berkeley Packet Filter (eBPF) codelets in the Linux kernel on millions of mission-critical servers around the globe. The code is automatically preempted if it runs longer than its predefined execution budget. It is also extremely fast, typically incurring less than 1 percent of overhead on the existing RAN operations.

Flexible software implementations

A software-based RAN solution enables easy modifications of the RAN software pipeline. It allows multiple instrumentation points in the pipeline that provide more fine-grained control and access to RAN processing and telemetry. Figure 4 below is a diagram showing a sample L1 software-based pipeline for massive MIMO, demonstrating examples of multiple instrumentation points in the pipeline.

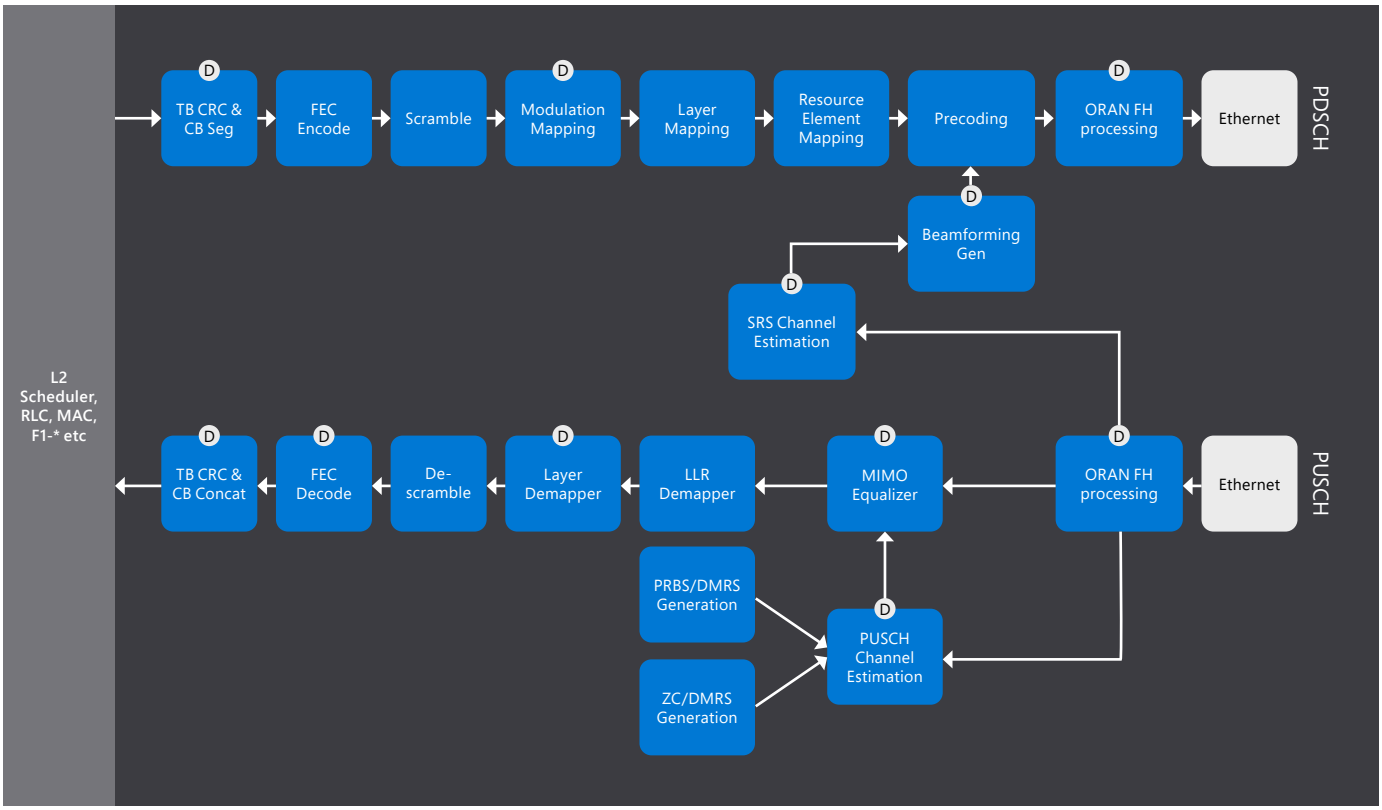

Figure 4: L1 software-based pipeline for Massive MIMO

These can be executed independently or combined for different use cases. Microsoft, Intel, and Capgemini have jointly prototyped the technology in Intel's FlexRAN™ reference software and Capgemini's 5G RAN. We also identified standard instrumentation points aligned with the standard 3GPP RAN architecture to provide maximum richness of information to external applications. Moreover, we have implemented the dynamic service models across these and showed that we could dynamically install code implementing a new service model in a live 5G RAN, operating at full capacity, without any performance impact. For more details about the technology, please visit the following link.

New applications of RAN and platform analytics and automation

The flexibility of the analytics and dynamic service models enables several exciting new applications, previously not thought possible. We illustrate them in the following examples.

RAN energy efficiency

The ever-increasing demand for network capacity makes cellular networks denser and more sophisticated and continues to push energy consumption and associated carbon footprint upwards. Thus, it is paramount to make 5G RAN, as one of the highest power-consuming parts of the 5G stack, very efficient. Vendors and operators have made good progress in this space by switching off radios operating at higher carrier frequencies during idle times and relying on lower frequencies to provide coverage. This approach has been further improved through baseband pooling to aggregate RAN workloads on fewer servers during less busy hours. Recently Intel has demonstrated an ability to further reduce vRAN power consumption by hibernating some of the CPU cores when not needed.

Most RAN energy-saving techniques are not very fast to deploy, and the energy-saving algorithms are typically operated at a diurnal timescale, following people's daily movements. Yet, even during peak hours, the network is rarely operated at full capacity due to the bursty nature of Internet traffic. There is a significant further potential energy gain if the traffic demand burstiness could be explored at faster time scales.

5G RAN is a carefully engineered real-time software. If a part of the system is in hibernation for energy savings, any instant traffic burst requires an immediate, sub-millisecond reaction to wake up more CPU cores. Otherwise, RAN performance can be seriously impacted, even leading to a crash. Therefore, any further innovation in RAN energy savings is hard to implement through external apps as they lack the full RAN visibility and telemetry required to predict and react to bursts at fast time scales. This limits developer innovation in one of the most important areas of O-RAN design.

Dynamic service models present an ideal solution to this challenge as they allow fine-grained access to many metrics across different layers of the RAN stack. Sending all those telemetries to a xApp would be prohibitively expensive in the volume of data. But with a dynamic service model, a developer can collect just the data it needs for a novel energy prediction algorithm, such as the number of active users, changes in different queue sizes, and more. Our initial prototype—built on top of Capgemini 5G RAN and Intel's FlexRAN™ reference software—can achieve up to 30 percent energy savings even during busy periods.

Anomaly detection

Virtualized RAN platforms are very complex. They require careful integration between various components, including hardware, operating system, timing clock, radio units, network switches, and RAN software. Various performance issues can occur due

to a misconfiguration of one or more components or other external factors. These are difficult to pinpoint, especially when different vendors provide different components with disparate logging and troubleshooting strategies. This situation is often labeled a lack of *single neck to choke* in the industry jargon and presents one of the biggest impediments to a broader O-RAN adoption.

Detailed platform and RAN telemetry can help mitigate these problems. With enough data and sophisticated AI/ML algorithms, many issues can be resolved automatically without explicit human involvement. For example, a sophisticated platform and RAN analytics and automation algorithm can correlate detailed telemetry from different platform components.

External interference detection is one such example. External wireless interference has long been a source of performance issues in cellular networks. It has become an even more pressing problem with illegal cellular repeaters readily available on Internet e-commerce sites. Detecting external wireless interference is difficult and often requires a truck roll with specialized equipment and experts to detect it. The problem is even more pronounced in enterprise 5G networks, where various machinery can cause interference leaks, and enterprise IT is ill-equipped and trained to deal with the problems.

With dynamic service models, we can turn an O-RAN 5G base station into a software-defined radio that can detect and characterize external wireless interference without affecting the radio performance. We have prototyped a dynamic service model that averages received IQ samples across frequency chunks and times inside a FlexRAN™ reference software L1 layer. It reports averages to an application that runs an AI/ML model for anomaly detection that can detect when the noise floor increases.

We perform this detection in a live 5G network. Due to the richness of the information available to the developers, our dynamic service model can identify and ignore 5G resource blocks with active transmissions and only looks for interference in idle slots. This is because dynamic service models can exchange information between themselves. For example, a dynamic service model with access to scheduler data can summarize which resource blocks are free while another dynamic service model with the access to IQ samples can collect samples from the free resource blocks. Such correlation of data collection between various stacks is extremely powerful.

By looking at the IQ samples from RAN Layer 1 only during idle slots (identified by a different service model (D) in Layer 2) and applying an interference detection algorithm, one can pinpoint performance issues related to external wireless interference. Similarly, various system logs can identify hardware faults across the system.

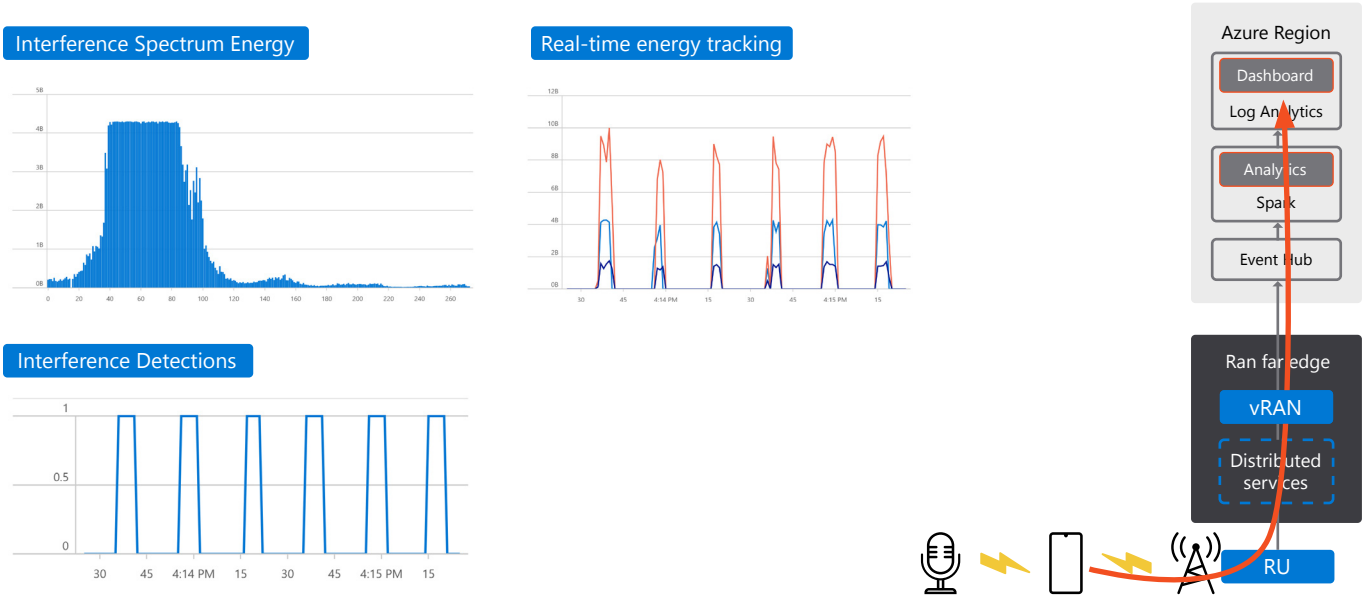

Figure 5: External wireless interference detection

Another example is a scenario where an OS update incorrectly places network interface card (NIC) interrupts onto the CPU core used by the RAN software. This can reduce RAN throughput since the RAN threads can be unexpectedly preempted. Such an issue is very difficult to detect. However, by correlating packet processing latency time series at different layers of the RAN stack with interrupt allocations at the OS level, a sophisticated AI/ML algorithm can easily detect an anomaly.

Today, detailed telemetry data is available from most platform components except RAN. However, with dynamic RAN service models, this vision can become a reality very soon.

Channel prediction

Efficient spectrum utilization depends on understanding wireless channel conditions such as signal strength, MIMO rank, and the number of retransmissions. Based on past channel performance and quality, modern AI/ML algorithms can accurately predict channel performance in the near future, potentially increasing spectrum utilization. The more information about the channel they can obtain and the quicker they can react, the higher spectral utilization can be achieved.

Dynamic service models help because they provide detailed, fine-grained channel information that can be fed to the channel prediction ML models. These lightweight models can be run at the far edge, providing channel estimates that can be immediately fed back to the rate adaptation and scheduling algorithms in the RAN, again through dynamic service models.

One demonstrated example of such an ML algorithm is Capgemini's Intelligent MAC scheduler. Here the ML model runs along with DU to predict the MCS to be used by the scheduler, improving spectral efficiency by 15 percent.

Customizing enterprise network slicing

Many enterprise 5G RAN applications require special customization. AR/VR, drone navigation, and coordination of a fleet of robots may require specific low latency guarantees. Other applications, such as video analytics, can be less latency sensitive but desire higher throughput. Complex use cases in different enterprise verticals may require customizing packet scheduling across different slices to achieve the right balance across applications.

Today, this customization is complicated and expensive. Specialized enterprise vendors must develop their own vertically integrated 5G RAN stacks to address these needs. Dynamic service models can expose the right level of telemetry and control over scheduling to allow innovators to implement novel scheduling algorithms in different slices that can be dynamically loaded onto a standard 5G RAN, reducing costs and speeding up innovation. The flexibility of software enabling low latency microservices leads the way to innovation as compute and communications merge.

Application to O-RAN architecture

The next generation of RICs will be required to support a wide range of new service models that have far stricter latency requirements than what is currently supported. To provide the low-latency needed for these new services, we proposed in the architecture described in previous sections the introduction of a new type of RIC, the real-time RIC (RT-RIC). As data and services continue to move at an ever-increasing pace, it is critical that we can support new services with the lowest possible latency. The RT-RIC enables this ability and therefore can become an essential component of the next-generation RIC architecture.

“As data and services continue to move at an ever-increasing pace, it is critical that we can support new services with the lowest possible latency.”

Conclusion

Virtualized RAN and O-RAN fundamentally transforms the way future radio networks will be operated. Open interfaces, intelligent radio controllers, and 3rd party applications promise a wave of innovation with the potential to reduce network operating costs and introduce new services. Current O-RAN interfaces and architectures are the first steps in this transformation. They demonstrate the vision and enable the first batch of innovative applications, mainly focused on mobility and traffic management.

In this white paper, we propose the next step in this transformation. By introducing dynamic service models that allow applications to access almost all available information and exercise control at different RAN layers, we offer the ability to unleash the full power of innovation on RAN transformation. These capabilities enable a truly open ecosystem where any small innovator with an impactful idea can quickly, easily, and safely deploy their innovation in O-RAN networks worldwide. A key enabler of this is the software implementation of the entire RAN stack so that all aspects can be analyzed to deliver best-in-class RAN.

Acronyms

AI	Artificial intelligence
API	Application programming interfaces
CNF	Containerized network function
COTS	Commercial off-the-shelf
C-RAN	Cloud RAN
CU-CP	Central unit control plane
CU-UP	Central unit user plane
D	Dynamics service model
D-RAN	Distributed RAN
DU	Distributed unit
eBPF	Extended berkeley packet filter
IQ	In-band and quadrature
L1	Layer 1
MIMO	Multi input/multi output
MAC	Medium access control
ML	Machine learning
ms	Millisecond
NIC	Network interface card
NFA	Network fabric automation
NFVi	Network functions virtualization infrastructure
nRT-RIC	Near-real-time remote intelligent communications
OFH	Open fronthaul
Open RAN	Open radio access network
O-RAN	Open radio access network
PHY	Physical layer
PRB	Physical resource block
RAN	Radio access network
RLC	Radio link control
RIC	RAN intelligent controller
RT-RIC	Real-Time RIC
SMO	Service management and orchestration
UPF	User plane function
VDU	Virtualized distribution unit
VNF	Virtual network function
vRAN	Virtual RAN