O-RAN next Generation Research Group (nGRG)

Research Report

Architecture principles for a cloud-friendly future 6G RAN architecture

Report ID: RR-2024-01

Contributors:

Ericsson

Nokia

III

Dell

Release date: 2024.02

Authors

Gunnar Mildh, Ericsson (Editor-in-Chief)

Elena Myhre, Ericsson

Hannu Flinck, Nokia

Christian Mannweiler, Nokia

Li Wan, III

Cheng Chang Chen, III

George Ericson, Dell

Reviewers

Hank Kafka, O-RAN Alliance

Bernard Guarino, O-RAN Alliance

Farooq Bari, AT&T

Gerd Zimmermann, Deutsche Telekom

Ravi Sinha, Jio

Disclaimer

The content of this document reflects the view of the authors listed above. It does not reflect the views of the O-RAN ALLIANCE as a community. The materials and information included in this document have been prepared or assembled by the above-mentioned authors, and are intended for informational purposes only. The above-mentioned authors shall have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of this document subject to any liability which is mandatory due to applicable law. The information in this document is provided ‘as is,' and no guarantee or warranty is given that the information is fit for any particular purpose.

Copyright

Executive summary

It is expected that the ongoing trend of cloud-based implementation and deployment of 5G network functions also will be highly relevant in the 6G/2030-time frame. This means that the future 6G RAN architecture should be designed with cloud-based implementation and deployment in mind, i.e., it should be cloud friendly.

The objective of this research report is to identify and analyze key architecture principles relevant to the standardization of a future cloud-friendly 6G RAN functional architecture. The identification and understanding of these principles are useful input to later 6G standardization activities in 3GPP and O-RAN.

<u>The architecture principles are arranged into 3 different areas:</u>

1. Principles related to future requirements on the 6G RAN architecture

These cover things like energy performance and sustainability, latency performance, observability, resiliency, and flexible deployment. Some example findings in this area include:

- Future RAN architecture needs to take significant steps in improving energy performance and HW usage. This includes efficient support of sleep mechanism across the network as well as means to pool HW enabling more efficient utilization as well improving energy efficiency.
- Resilience is seen as important across many areas including compute, storage and networking. This includes also designing the network applications to be able to tolerate failures in the underlying infrastructure and continue to operate. It is expected that most of this will be addressed in implementation and deployment, however it is likely that some standard enablers are needed similar to NF set defined in 3GPP etc. [4].
- Latency performance is seen as important to enable future high-quality, real-time applications. It is important to design the cloud-friendly architecture with this in mind. This can include special consideration when utilizing cloud mechanism for service chaining, utilizing shared cloud platforms, and modularizing control and user plane function so that this does not add unnecessary latency to end user applications.

2. Principles related to future standardization of the 6G RAN architecture

These cover ways to achieve multi-vendor interoperability while still allowing innovation in implementation and deployment, ways to make 6G RAN protocols more cloud-friendly, and achieve good separation of concerns (i.e., avoiding unnecessary dependencies) between NFs and layers.

Some example findings in this area include:

- Ensure good separation of concern across different multi-vendor interfaces and layers. This includes solutions to minimize exposure of NF implementation specific information, allowing larger changes to a NF without impacting other NFs. It also includes allocation of functionality to NFs avoiding that a certain functionality is split between different NFs creating unnecessary dependencies and signaling impacting complexity and latency.
- Importance of adopting cloud-friendly protocols within future RAN architecture. This can include signaling transport and security protocols which are well supported on current and

future cloud platforms. It also includes clear separation of signaling application protocols from underlying signaling transport protocols enabling independent evolution of these layers.

3. Principles related to future deployment and management of the 6G RAN architecture

These cover things like general principles for automation, automated root cause analysis, zero trust architecture, transition towards more DevOps, and ways to optimize state handling and utilize data-meshes.

Some example findings in this area include:

- Designing for automation including support for closed loop automation to improve network performance and enabling continuous service assurance. Intent based APIs, simplifying complex RAN configuration, and enabling quick deployment of new services. Improved observability to better understand end user quality of experience and impacts of configuration changes. Support for Life Cycle Management (LCM) and orchestration.
- Support for automated Root Cause Analysis (RCA), enabling identification of underlying causes of problems, incidents, or failures within a system or process. This will be critical in a cloudified RAN since the network will be built with components from different vendors and include independently designed and managed HW and SW. The standard should focus on building enablers for automated RCA.
- Supporting Zero Trust Architecture (ZTA) with fine-grained authentication and authorization of NF interactions based on application and platform state, regardless of the requester's location in the network. ZTA is particularly important in a multi-stakeholder cloud setting. Mechanism such as Trusted Execution Environments (TEE) and remote attestation can be enablers for protecting resources and monitor the security of the application and platform.

Table of Contents

Authors 2

Disclaimer 2

Copyright 2

Executive summary 3

List of abbreviations 6

1 Background 8

2 Objective and scope of this research report 8

3 What is a cloud-friendly RAN architecture? 8

4 Proposed architecture principles 9

4.1 Requirements on 6G RAN architecture 9

4.1.1 Energy Performance and sustainability 9

4.1.2 Latency Performance for Future Applications 9

4.1.3 Observability 12

4.1.4 Resiliency 13

4.1.5 Flexible deployment 14

4.2 Standardization related principles 15

4.2.1 Holistic approach when introducing new functionality 15

4.2.2 Cloud-friendly signaling protocols 15

4.2.3 Minimizing inter-function and layer dependencies 16

4.3 Deployment and management related principles 17

4.3.1 Design for automation 17

4.3.2 Support for automated Root Cause Analysis (RCA) 18

4.3.3 Supporting a Zero Trust Architecture 19

4.3.4 Rapid DevOps-like development cycle 20

4.3.5 State-optimized and data-mesh-ready RAN design 21

5 Conclusion 22

References 23

List of abbreviations

AEAD	Authenticated Encryption with Associated Data
AI/ML	Artificial Intelligence / Machine Learning
API	Application Programming Interface
CI/CD	Continuous Integration / Continuous Delivery
CN	Core Network
CNF	Cloud native Network Function
CPU	Central Processing Unit
C-RAN	Centralized RAN
CU	Central Unit
CU-CP	Central Unit Control Plane
CU-UP	Central Unit User Plane
DU	Distributed Unit
E1AP	E1 Application Protocol
EAP	Extensible Authentication Protocol
FFS	For Further Studies
eMBB	Enhanced Mobile Broadband
HTTP	Hyper Text Transfer Protocol
HW	Hardware
I/O	Input / Output
L1	Layer 1
LCM	Life Cycle Management
mMTC	Massive Machine Type Communication
NAT	Network Address Translation
NF	Network Function
NGAP	Next-Generation Application Protocol
OS	Operating System

PNF	Physical Network Function
QoS	Quality of Service
RAN	Radio Access Network
RCA	Root Cause Analysis
RLF	Radio Link Failure
RRC	Radio Resource Control (protocol)
SCTP	Stream Control Transmission Protocol
SLA	Service Layer Agreement
SW	Software
TCP/IP	Transmission Control Protocol/Internet Protocol
TEE	Trusted Execution Environment
URLCC	Ultra-Reliable Low Latency Communications
VNF	Virtual Network Function
ZTA	Zero Trust Architecture

1 Background

There is an ongoing evolution in the telecommunications industry related to how networks are being implemented, deployed, and operated, including, but not limited to, cloud-based implementation and deployment of 5G network functions in RAN and CN. It is expected that this trend of cloud-based implementation and deployment will continue also in the 6G/2030-time frame. This means that the future 6G RAN architecture should be designed with cloud-based implementation and deployment in mind, i.e., it should be cloud friendly.

2 Objective and scope of this research report

The objective of this research report is to identify and analyze key architecture principles relevant to the standardization of a future cloud-friendly 6G RAN functional architecture. The identification and understanding of these principles are useful input to later 6G standardization activities in 3GPP and O-RAN. The goal of the research report is to give guidance on how (and how not) the 6G RAN architecture can be designed with regards to cloud friendliness.

Note: The research on 6G is in an early phase and the final 6G architecture will need to fulfill many requirements (incl. legacy inter-working and migration) so any recommendation in this report should be considered with this in mind.

3 What is a cloud-friendly RAN architecture?

The mobile communications networks are transitioning from tightly integrated SW and HW solutions into SW-based and cloud-deployed solutions. This transition has been enabled (and fueled) by the emergence of cloud and cloud-native technologies. These technologies are currently evolving at a rapid pace, which is expected to continue. Consequently, it has become important to ensure that the architecture of the mobile network is defined such that it allows making use of evolved and new emerging technologies and their potential benefits, such as simplified management and orchestration, reduced total cost of ownership (TCO), or higher scalability, when implemented. A RAN architecture incorporating these capabilities is in this report referred to as a cloud-friendly RAN architecture.

Even though cloud friendliness is desired for the future RAN, some care must be taken not to compromise (too much) on performance. E.g., time critical procedures may have to be optimized with focus on performance thereby possibly sacrificing some of the cloud friendliness capabilities. Also, the need to optimize radio resource utilization, as mentioned below, may conflict with cloud friendliness.

RAN functionality and deployments should be designed and planned to intelligently optimize the use of the precious and constrained radio resources and support the highly diverse services (e.g., evolved mobile broadband, immersive communication and extended reality, digital twinning, low power wide area access). The strictest requirements on RAN performance are on latency (incl. latency variation), throughput and reliability. These requirements in addition to significant L1 processing drive the need to support complex digital signal processing. Such processing is not a typical application for cloud implementation.

Therefore, cloud-friendly RAN architecture and RAN friendly cloud need to be considered both together. The RAN architecture needs to be able to leverage flexibility, scalability, agility, expedition of service development and customization with CI/CD, etc. enabled by the cloud technology. The cloud should be designed to support the needs to the RAN, e.g., real-time operating system and preemptive scheduling to ensure deterministic response time for the time-sensitive events and tasks, a large amount of CPU cycles and memory to support high-performance RAN applications requirements, guarantee resources, support the complex processing at the RAN by integrating hardware accelerators.

4 Architecture principles

4.1 Requirements on 6G RAN architecture

4.1.1 Energy Performance and sustainability

Description and motivation

Energy performance and HW utilization are key performance metrics of future sustainable networks. Several aspects are of importance including low idle mode power consumption (allowing to switch off HW during times of excess capacity), efficient processing at high load (utilizing state of the art SW/HW components), and efficient pooling and HW reuse [1].

Examples illustrating the principle

The RAN architecture should support solutions utilizing micro-sleep periods, as well as solution to switching off extra processing, or cell/frequency related HW during periods of low network load. It should also support Centralized RAN (C-RAN) deployments with pooled HW over a large area thus allowing the network to more be optimized for average load, rather than peak load.

Impact to functional architecture (future standard)

The 6G system needs to support functionality to efficiently switch off and pool HW, as well as the efficient utilization of HW acceleration. It is For Further Studies (FFS) how this will impact the functional architecture. It is however expected that 6G radio interface and RAN should be designed with this in mind to further improve the network energy performance compared to 5G. The radio access should support energy and processing efficient AEAD modes (encryption combined with integrity protection) based on algorithms that can perform well in both hardware and software (utilizing widely available CPU based acceleration). Efficiency and state-of-the-art algorithm support in relevant deployment models should also be a consideration in choosing security protocols for the network domain security.

4.1.2 Latency Performance for Future Applications

Description and motivation

When designing a cloud-friendly 6G RAN architecture, a primary objective is to ensure seamless, high-quality, reliable, and real-time communication for heterogeneous and diverse applications.

In essence, latency consideration is vital in the future cloud-friendly 6G RAN design to guarantee superior user experiences, support mission-critical applications, optimize network resources, and future-proof the network for evolving technological demands.

Enhanced user experience relies on quick response times. Latency directly affects the perceived quality and responsiveness of services. Mission-critical applications like autonomous driving and telemedicine require ultra-reliable low-latency communication (URLLC) for safety and effectiveness. Efficiently addressing latency can improve network resource utilization, reducing costs and enhancing performance. With emerging technologies demanding faster response times, a low latency RAN architecture ensures the network remains relevant and competitive.

So, when designing future cloud-friendly 6G RAN architecture, the significant factors affecting latency, such as Dynamic Service Chaining, Virtualized shared OS, deployment strategy, functional grouping, etc., need to be considered and analyzed.

Examples illustrating the principle

a) Dynamic Service chaining involves linking multiple network functions (NFs) together to craft a service tailored for users or applications. These NFs can either be software or hardware-based, undertaking various roles including Base Station, Core Network, RAN Intelligent Controller, load balancing, firewall, encryption, and more. A cloud-friendly RAN architecture, which separates the control plane from the data plane and enables distributed, programmable network control, provides an environment conducive to dynamic service chaining. It does so by offering flexible and streamlined allocation and orchestration of NFs across various network nodes. Yet, integrating dynamic service chaining within the cloud-friendly 6G RAN architecture presents challenges in guaranteeing deterministic Quality of Service (QoS). This QoS is characterized by delivering a consistent, reliable performance level based on metrics such as delay, jitter, throughput, and packet loss. Factors influencing the QoS in this architecture include the exchange of signaling messages within the control plane and packet transmissions in the data plane. Consequently, crafting mechanisms to reduce latency and optimize efficiency across both planes is of paramount importance.

b) Virtualization in cloud architecture offers numerous advantages, including flexibility, scalability, and enhanced resource utilization. However, it also has inherent overheads that can affect both performance and latency. The latency in such an architecture can be attributed to several factors:

- Hypervisor Overhead: Hypervisors introduce an intermediate layer between the physical hardware and the virtual OS. Delays might occur, especially when the hypervisor has to translate and forward a large number of instructions from the virtual function to the physical hardware.

- Resource Contention: In a virtualized environment, multiple virtual functions share standard physical resources such as the CPU, memory, and I/O. When these functions try to access resources simultaneously, it can lead to resource contention. This contention can cause queuing delays, particularly if a virtual function is resource-intensive or if there is overprovisioning of the foundational physical resource.

- I/O Virtualization Overhead: I/O operations in virtualized environments typically involve an extra translation step. The combination of this translation with the management of I/O requests creates an overhead, which can increase latency, especially in data-intensive tasks.

c) Deployment strategy causes a latency concern, especially when services are deployed in multiple regions, using microservices, or connecting to hybrid cloud/on-premises environments. There are various deployment strategies, such as Single Region Deployment, Multi-region Deployment, Microservices Deployment, Serverless Deployment, Hybrid Cloud/On-Premises Deployment, and more. Multiple factors and components come into play. Here's a breakdown of the sources of latency and considerations to keep in mind:

- Network Latency: The physical distance between services can introduce significant latency, especially if they're in different regions. Additionally, the number of hops (routers, switches, ISPs) that data or messages must pass through can introduce variability in latency. Moreover, limited bandwidth can cause congestion, especially during peak traffic periods.
- Encryption/Decryption Overhead: Data transferred between services typically undergoes encryption. The process of encrypting/decrypting can introduce latency, especially if it's not hardware-accelerated.
- Data Transfer and Synchronization: Real-time synchronization of data between services can lead to increased latency, particularly if large amounts of data need frequent synchronization.
- Integration Overhead: The complexity of integrating 6G RAN services with cloud services might add additional points of failure or latency, especially if involving middleware or gateways.
- Service/API Call Latency: When one service makes API calls to other cloud-friendly services (or vice versa), latency might be introduced, depending on the responsiveness of those services and the network.
- Inconsistent Environments: Discrepancies in configuration, hardware, or software versions between service deployment environments can lead to unpredictable performance and latency.
- Cold Starts in Cloud Services: Some cloud-friendly services, especially serverless ones, experience cold start latency, where the first request after a period of inactivity takes longer.

d) Similar to Service Chaining in the User Plane (e.g., [2]), disaggregation of Network Functions in the control plane also has a performance impact on latency. There are many time-critical RAN control procedures, such as RRC reconfiguration, connection establishment or resume. Delays in these procedures may result in data or radio link loss which could have a negative impact on end user performance. If such a procedure requires signaling and coordination between many NFs, its execution time will increase by the overhead represented by the formulation and encoding of the messages, their transmission, reception, queuing and decoding. Thus, it is desirable to reduce the number of NFs the required functionality is spread across.

Impact to functional architecture (future standard)

In the quest to optimize latency performance for future cloud-friendly 6G RAN architectures, it's imperative to identify and analyze the crucial factors that influence latency within the architectural design. This approach guarantees a design that not only offers robust support for emerging applications but also upholds structural integrity and functionality.

As we contemplate the future of cloud-friendly 6G RAN architecture, the critical role of latency performance for upcoming applications becomes clear. It's not merely about minimizing delays; it's about crafting a well-defined architecture capable of seamlessly supporting real-time applications, augmented reality experiences, remote surgeries, and other advanced services. When we delve into the topic of latency, it naturally raises an important question: how will this emphasis reshape the functional architecture? Can traditional hierarchies and layers meet the demands, or will we witness a shift towards more decentralized or edge-centric designs? These considerations highlight the interplay between performance metrics and architectural dynamics, emphasizing the need for a holistic approach in design and planning.

4.1.3 Observability

Description and motivation

Observability is the practice of gaining insight into the internal behavior of a system by collecting, analyzing, and visualizing relevant data. It refers to the ability to understand and reason about the system's state, performance, and behavior based on the available data. Observability focuses on providing actionable insights into complex distributed systems, allowing operators and developers to diagnose and troubleshoot issues effectively.

The motivation behind observability is to address the challenges posed by complex and dynamic systems, such as microservices architectures and cloud-native applications from different vendors. Traditional monitoring approaches often rely on predefined metrics and alerts assuming a predefined structure between the system functionalities. This assumption may not provide sufficient visibility into the system's internal dynamics. Observability in cloud environment aims to provide a holistic view of the dynamics of heterogenous systems composed of functionality from multiple vendors. It enables to understand and agree on how responsibilities are distributed among interaction system components in functional scope.

Examples illustrating the principle

a) Logs: Observability relies on collecting and analyzing logs generated by various components within the system. Logs capture information about events, errors, and activities, providing a historical record of the system's behavior. Logs can be aggregated, indexed, and searched to identify patterns, detect anomalies, and gain insights into the system's performance.

b) Metrics: Observability involves capturing and analyzing metrics, which are quantitative measurements of system behavior. Metrics can include CPU usage, memory consumption, network traffic, (radio) resource utilization, response times, and more. By monitoring and analyzing metrics, teams can identify trends, performance bottlenecks, and abnormalities in the system's behavior.

c) Tracing: Tracing allows for the tracking and visualization of the flow of requests across different components and services within a system. Traces provide a detailed view of the path a request takes, including latency and time spent at each step. Tracing helps identify performance issues, bottlenecks, and dependencies between components.

d) Distributed Tracing: In complex distributed systems, observability often involves distributed tracing, which tracks requests as they traverse multiple services and components. Distributed tracing provides a comprehensive view of the interactions and dependencies between various services, helping identify performance issues, latency hotspots, and service dependencies.

e) Dashboards and Visualization: Observability involves presenting collected data in a meaningful and visual manner. Dashboards and visualization tools provide real-time and historical views of the system's health, performance, and behavior. They enable teams to monitor key metrics, detect anomalies, and correlate data from different sources for comprehensive analysis.

Impact to functional architecture (future standard)

6G standards should increasingly consider avoiding specifying observability procedures that are deeply integrated or even dependent on the functional architecture of the 6G system. Rather, the functional architecture should be defined in a way that it embraces multiple observability procedures, as favored by the chosen implementation and also deployment option.

Moreover, fundamental observability procedures should also enable interoperability between different NFs as well as between NFs and the cloud infrastructure, thus facilitating observability across a complex multi-vendor system. With respect to current O-RAN architecture, this may result in additional requirements for O1 and O2 interfaces. Procedures that could build on top of such fundamental observability procedures include, but are not limited to, proactive monitoring and issue resolution, root cause analysis and troubleshooting, performance optimization, dependency management, and SLA management.

4.1.4 Resiliency

Description and motivation

Resiliency is required to ensure service availability in failure cases. It is the ability to "provide and maintain an acceptable level of service in the face of faults and challenges to normal operation" [3]. In a cloud environment, resiliency needs to cover software execution, computing, storage and networking functionality which may all contribute to service scalability, performance, and availability. Cloud native Network Function (CNF) and Virtual Network Function (VNF)-based RAN environments differ from an ordinary cloud environment in that not everything can be virtualized because Physical Network Functions (PNFs) are also involved. The NFs and applications should be designed so that they can tolerate failures in the underlying infrastructure independently of the applied resiliency and recovery mechanisms.

Examples illustrating the principle

a) Redundant service instances can ensure service continuity and improve performance but with an increasing overall cost overhead. Depending on the service and its distribution across the infrastructure different degrees of redundancy can be supported.

b) Scalability: The processing capacity of individual RAN functions (e.g., of CU-UP) should be dynamically adapted to match the available capabilities of the underlying infrastructure, selecting among multiple options (such as vertical scaling, where more capacity is added for a service within in a node, or horizontal scaling, where new nodes are providing resources for service such as NF migration, re-directing of traffic, etc.).

c) CNF and VNF resilience can be achieved by optimal resource orchestration of multiple CNF/VNF instances across multiple regions and clouds. However, resilience of the PNFs relies more on strategic network planning and fault prediction because they cannot leverage similar reactive on-demand horizontal scaling as CNFs and VNFs. However, common between CNFs, VNFs and PNFs is the need for effective application-level mechanisms to handle failures, together with an efficient monitoring and data collection capability and resource planning.

d) Load balancing: Resilience is increased by appropriate load balancing schemes, e.g., for user plane traffic using pre-defined criteria, (e.g., association between CU-UP and DU), thus also relieving the NF developer from including this task in the "functional logic" of the NF.

Impact to functional architecture (future standard)

The architecture should mandate monitoring, performance data collection, root cause analysis, fault forecasting, fault avoidance, and conflict resolution functionality. Out of service, recovery times, failure rate times should be specified. However, how these functionalities are to be realized should be left for implementation. Compatibility with 3GPP resiliency approaches, e.g., NF Set and NF Service Set Clause 5.21.3 [4], should be investigated.

4.1.5 Flexible deployment

Description and motivation:

Cloud deployment can be used to increase the flexibility of deployment of different network and service layer functions due to the decoupling of SW and HW. From an operator point of view this could make it easier and quicker to deploy new functionality in the network, and it can also be used to support specific use cases such as on-premises deployment for verticals.

As discussed in 4.2.1 it is important that 6G standardization does not limit this deployment flexibility but rather that it can work as an enabler.

Examples illustrating the principle:

a) 6G standardization should provide enablers allowing the optimal deployments and selection of different NF instances, e.g., considering mobility, load, transport network, service requirement.

b) 6G standardization should allow deployment on private and public clouds (or a mix of them), bare metal or virtualized environments (or a mix of them), as well as a range of CPUs and computing architectures and in a wide range of geographical arrangements.

c) One possible area to further explore for cloud-friendly RAN architecture is to support RAN NFs spanning over multiple network sites, i.e. similar to NF sets [4]. This would allow geographically redundant operation for these RAN NFs without impacting other NFs.

d) 6G standardization should enable the network to be deployed in a wide variety of ownership models (sharing, venue-owned, tower companies, neutral host, etc.) and scenarios (ad-hoc network extensions and boosts, emergency networks, digital airborne communications, etc.) For example, there may be some restrictions on the possibility to have a common service repository or service discovery system for both the RAN and the CN if they are under different ownership.

Impact to functional architecture (future standard):

To allow flexible deployments of cloud-friendly RAN functionality careful consideration is needed with regards to how the architecture incl. management is standardized. The functional architecture may leave certain details open for the implementation and deployment architectures to cover, coupled with a richer management and automation specification to handle the increased flexibility. Security frameworks must be able to dynamically support protection and authorization for new functionality deployed.

4.2 Standardization related principles

4.2.1 Holistic approach when introducing new functionality

Description and motivation

Cloud based implementation and deployment is characterized in concepts such as independent scaling micro-services, CI/CD, load balancers, reliable databases, etc. Not all of this is however suitable to be standardized. The main goal of standardization is to support multi-vendor deployments, while still giving freedom to innovate and benefit from state-of-the-art tools in the implementation and deployment. As such it is important to consider the network as a whole (incl. standard, implementation, deployment) when discussing how new functionality should be supported since many challenges can be addressed with a combination of standard, implementation and deployment features.

Examples illustrating the principle

For high reliability and availability, it is possible to utilize a combination of functionality addressed at different layers or domains:

a) Redundant sites, transport links, extra HW etc. (deployment)
b) Fast failover, internal N + M redundancy, reliable data bases (implementation)
c) Procedure for radio link re-establishment at RLF (standard)

Impact to functional architecture (future standard)

When studying how to support new functionality in future 6G RAN architecture it is important to consider what functionality needs to be standardized, and what functionality can best be addressed in the implementation and deployment. It is important that 6G standardization does not introduce constraints restricting the possibility to innovate in the implementation and deployment.

4.2.2 Cloud-friendly signaling protocols

Description and Motivation:

First, it is assumed that the 6G system, based on the principle of minimizing inter function dependencies as in 4.2.3, will have the ‘right’ interfaces specified, where the functional split is clear and useful and there is a good possibility to deploy and integrate in a multivendor deployment.

With the above in mind, to enable the benefits of cloud systems, network internal signaling protocols also need to be designed and specified so they assume a cloud-based infrastructure as the fundamental basis for implementing those protocols.

Application layer signaling entities (such as NFs, xApps, rApps) and underlying signaling transport should therefore be adapted to cloud principles, including using state-of-the-art security mechanisms, efficient support for load-balancing, and be future proof e.g. build on abstractions of underlying layers, high-level enough to allow the layers to evolve independently.

Moreover, the focus should be on e2e communication between relevant entities e.g. NFs, avoiding complicated stateful proxies and building on the assumption that we will still have layering and separation between signaling application protocols (e.g. such as NGAP [5], E1AP [6] defined in 3GPP) and signaling transport (e.g. SCTP [7] over IP, HTTP over TCP/IP) .

Examples illustrating the principles:

The current signaling transport within RAN and between RAN and CN in 5G is based on SCTP. For 6G it should be considered to replace SCTP, which is not commonly used in cloud deployments outside telecom (due to poor NAT support, kernel-based stacks, small community, slow evolution) with something more cloud friendly, assuming this is feasible from a migration perspective. In addition, the functional split and coupling between the signaling application and transport should be investigated to ensure good support for e2e communication between signaling applications, flexible load-balancing, scaling, etc. and at the same time avoiding the application behavior being dependent on specific signaling transport functions. The security solution for application-level protection (e.g., mutual authentication) will depend on how the transport protocol evolves going into 6G.

Impact to functional architecture (future standard):

As already stated above, for signaling interfaces within RAN, we may want to choose different transport protocols for example, not choosing SCTP), and potentially move some services currently in the transport to the signaling application layer. This should be carefully studied, including possible alternatives and how the application protocols may be designed to be able to cope with multiple options for the underlying infrastructure. This would enable the underlying infrastructure to evolve independently without jeopardizing the above functionality. Depending on how the transport protocols evolve, a well-reviewed security solution that is efficient in relevant deployment models should be supported in the standards for application-level protection on the relevant interfaces.

4.2.3 Minimizing inter-function and layer dependencies

Description and Motivation:

A fundamental principle for good architecture design is to try to maximize the separation of concerns between different layers and logical entities such as NFs. Separation of concerns include multiple aspects such as:

a) minimizing implementations specific information needed in NF A about NF B, thus making it possible to make larger changes to NF B while still interworking with NF A

b) reduce functional dependencies between NFs to allow innovation in the implementation of an NF to optimize the functionality of that NF. This could for instance include allocating the responsibility of a certain set of functions to a single NF (i.e. avoid that the responsibility is split by multiple NFs)

c) concentrate the related context information to a single NF to minimize the frequent inter-NF signaling.

d) ensure separation in NF design considering the different performance, processing and platform requirements of network functions and their offered services.

e) designing a layered system such that the functions of a layer below are defined, but not dependent on how the functions are realized.

Although the principle above is applicable for any architecture it becomes extra important in a cloud-based deployment due to the desire to allow independent development and deployment and, ultimately, high feature velocity. With good separation of concerns, it should be possible to support flexible deployment and to add new functionality to already deployed NFs and optimize existing functionality without impacting other NFs and the standard. Separation of concern can also have security benefits since if less interactions are needed between different NFs less sensitive information are also exchanged which improves security and could make it possible to simplify the security architecture.

Examples illustrating the principles:

In current 5G network there are examples of successful separation of concerns (at least with regards to multi-vendor deployments) such as the split between CN and RAN with relatively clear separation of responsibilities, as well as less successful examples, such as the split between CU and DU with regards to serving and secondary cell selection where the responsibility for the UE configuration, constrained by the UE radio access capabilities, is shared between the CU and DU.

Other examples of successful separation of concerns include the HTTP protocol that underwent a series of evolution steps replacing the underlying technologies targeting higher performance and flexibility with HTTP/3 without impacting the user of the protocol. Another example is service-oriented architectures, where the focus is on the services, but exact way of providing the service is left unspecified allowing better implementations. Yet another example is the Extensible Authentication Protocol (EAP) protocol allowing easy addition of new authentication schemes.

Impact to functional architecture (future standard):

Separation of concerns should be considered carefully when specifying the functional architecture of 6G. As in previous generation it is expected that there will be a need for a number of multi-vendor interfaces, and when specifying these multi-vendor interfaces, it is important to have a clear separation of concerns and minimize inter-NF dependencies. In some cases, this means that dependent functionalities, or functionalities that operate on the same content, should be bundled in the same NF.

4.3 Deployment and management related principles

4.3.1 Design for automation

Description and motivation

Automation enables managing ever increasing system complexity while supporting resiliency and improving process quality since it applies deployment, configuration, optimization, repair, and failure recovery actions in a faster and more consistent manner than humans. Since automation intersects network services, orchestration and the cloud platform interoperable interfaces are essential to seamless operations. Key enabling features for automation include observability, analytics and AI/ML tool set, intent-based management interfaces and closed-loop operations [8]. The architecture should support full automation of network and service life cycle management to optimize the system performance with minimal human interaction and to simplify network configuration and operations. With the help of Digital Twins, the impact of candidate configurations can be tested without affecting the operational network. As AI/ML is used throughout the RAN incl. automation, trustworthy AI/ML (i.e., AI/ML that is safe, robust, explainable, privacy-preserving) becomes important to ensuring privacy and security in the system. Trustworthy AI/ML will likely not only be achieved through technology, but also through a better understanding of the limitations of the technology and how it affects security and privacy in a given application.

Examples illustrating the principle

a) Closed loop operations enable continuous network optimization and efficient resource usage to achieve service assurance & fulfillment targets. Multiple closed loops can run simultaneously, and they need to be coordinated to avoid suboptimal or even contradicting outcomes and actions. Also, a robust root cause analysis engine with fault management, performance change management and configuration management are needed.

b) With Intent based configuration APIs, the complex RAN configuration can be simplified by providing the target state of the system and leaving the details to the next lower level to provisioning.

c) AI/ML tools aim at improving closed-loop autonomous decision making in operations at subsystem, and system levels. Explainability of AI/ML can be critical to, for example, root cause analysis of incidents and errors.

d) Autonomous decision-making mechanisms can be bounded by rules and policies to satisfy the operational conditions under which autonomous operation is allowed.

e) Observability: The impact of a configuration change should be measurable and relevant performance data should be exposed to the rest of the system(s).

f) Automation supported onboarding & Lifecycle Management (LCM): this includes the CNF onboarding (DU & CU) on O-cloud, include initiation, scaling, healing, termination.

Impact to functional architecture (future standard)

A thorough analysis regarding potential dependencies between 6G RAN functions and proposed automation mechanisms is required before considering any standards impact. 6G standardization needs to support interfaces that enable automation tasks as well as mechanism to get real-time notifications of events. Extensions to ongoing efforts to intent-based APIs are also likely. These impacts are applicable both to cloud and non-cloud RAN deployment.

4.3.2 Support for automated Root Cause Analysis (RCA)

Description and motivation

Root Cause Analysis (RCA) is a systematic approach used to identify the underlying causes of problems, incidents, or failures within a system or process. It aims to go beyond addressing the immediate symptoms and instead focuses on understanding the fundamental reasons that contribute to the occurrence of the problem. The motivation behind RCA is to identify and address the root causes to prevent the recurrence of similar issues in the future, improve system reliability, and enhance overall performance.

In large-scale cloud infrastructure operations, RCA is an essential component of Incident Management, where it helps to investigate and resolve incidents. When an incident occurs, RCA involves a thorough examination of the event, including analyzing logs, system data, and user reports. The goal is to determine the underlying factors that led to the incident. By identifying the root causes, appropriate corrective actions can be taken to prevent similar incidents from happening again.

In cloudified RANs, automated RCA becomes an increasingly challenging task since such networks may be built (at least partially) with components from different vendors, incl. cloud infrastructure suppliers. A cloud -friendly RAN could hence re-use and integrate with the RCA mechanisms for cloud infrastructure management and extend them to coordinate with RAN-specific functionality and automated fault management. Finally, automated RCA can also be important to security monitoring and understanding malicious faults and incidents.

Examples illustrating the principle

a) Fault Management and Diagnostics: RCA plays a crucial role in fault management and diagnostics. It enables a systematic approach to identifying the root causes of network failures, disruptions, or performance issues. By conducting RCA, operators can determine the underlying factors contributing to the faults, such as hardware failures, software bugs, configuration errors, or network congestion. This knowledge enables efficient troubleshooting, timely resolution, and the implementation of preventive measures.

b) Network Planning and Optimization: RCA informs network planning and optimization activities. By analyzing historical performance data and conducting RCA on network-related issues, operators can identify areas for improvement. This includes addressing coverage gaps, capacity bottlenecks, radio frequency interference, or network congestion. RCA helps operators make informed decisions regarding network expansion, equipment upgrades, and optimization strategies. In cloud RAN deployments, these RAN-specific RCA methods need to be integrated or at least coordinate with traditional cloud infrastructure RCA processes.

c) Service Quality Improvement: RCA can drive service quality improvement. By identifying the root causes of service degradations or customer complaints, operators can take corrective actions to enhance service delivery. RCA helps in identifying issues related to network capacity, coverage, signaling, or QoS parameters. By addressing these root causes, operators can optimize network performance, minimize service disruptions, and improve customer satisfaction.

d) Continuous Improvement and Lessons Learned: RCA contributes to continuous improvement and knowledge sharing. By conducting RCA after significant incidents or failures, operators can extract valuable insights and lessons learned. These findings can be shared across the organization and industry to prevent similar issues in the future. RCA encourages a culture of learning, proactive problem.

Impact to functional architecture (future standard)

Standards should focus on building enablers for automated Root Cause Analysis (RCA), e.g., baseline means for components interoperability via standardized interfaces and data formats for important RCA use cases. This facilitates automated recovery even in multi-vendor environments and under severe outage conditions.

4.3.3 Supporting a Zero Trust Architecture

Description and motivation

A Zero Trust Architecture (ZTA) [9] incorporates a fine-grained authentication and authorization in evaluating resource requests (such as NF interactions) based on application and platform state, regardless of the requester’s location in the network. The idea that entities should not be trusted by default also extends to – when possible – minimizing implicit trust between layers, and strengthening solutions to still provide some security even after a potential compromise. This can be especially relevant in a multi-stakeholder cloud setting. Trusted Execution Environments and remote attestation can be technology enablers by providing ways to protect resources and make it possible to obtain secure measurements about application and platform state.

Examples illustrating the principle

a) Integration with Trusted Execution Environment technology can for example protect intellectual property and cryptographic keys in applications running on a cloud infrastructure. It can also provide a secure and isolated environment for slices.

b) Integration with remote attestation can augment authorization between 6G NFs with continuous checks on application and platform integrity.

c) Authentication and authorization at the application level between NFs provides a means to limit reliance on protection mechanisms external to the NFs. At the same time, A ZTA will continue to leverage perimeter protection and network segmentation as used in today's networks to, for example, protect the availability of services.

Impact to functional architecture (future standard)

Specifications should support a trust orchestration where a cloud infrastructure can be verified as trustworthy and on which NFs can then be securely deployed and automatically orchestrated with secure identities and configuration. Differences between different TEE technologies should be absorbed in more generic APIs in such a way that, for example, NFs can leverage information from remote attestation without necessarily performing it themselves. RAN specifications should support application-level cryptographic protection between NFs on all multivendor interfaces.

4.3.4 Rapid DevOps-like development cycle

Description and Motivation

One driver for cloud deployment is to reduce the time to market for new features and updates by decoupling the SW from the underlying HW. To facilitate fast innovation and rapid optimization the IT industry has arrived at the DevOps model, where the developer of a component is also responsible for operating it. (This relies on the principle of minimizing inter function dependencies 4.2.3) This allows the developer to learn about every aspect of its component and makes rapid feature upgrades and optimizations possible. The full DevOps model is possible for organizations developing their own application in-house. Since the telecom industry is structured differently, more collaboration between vendors and operators is required to allow some of that agility.

Most industry definitions agree that cloud native is about speed and agility [10]. To reach such agility. not only the application (and its functional architecture) needs to be designed well, but certain operational practices must also be followed as outlined above. Such operational practices need supporting technologies (for example, CI/CD pipelines, data collection and feedback mechanisms) that would benefit from industry standards. Best security practices such as secure coding, supply chain security and vulnerability tracking must also be integrated into the rapid development cycle.

Examples illustrating the principles

a) Vendors need the ability to frequently deliver software into a staging area of the operator.

b) Collaboration on integration and verification of the delivered software with other systems of the operator is required.

c) Operational data should be fed back from the operator to the vendor in a controlled, anonymized manner to allow timely software improvements. This requires a data infrastructure that allows the operator to provide selected operational data back to the vendors.

Impact to functional architecture (future standard)

A generic architecture for sharing data from NFs back to the vendor should be studied. Some parts might be standardized, some parts may use existing IT frameworks, and some parts should not be standardized to give freedom to operator to integrate to its existing systems. Such data infrastructure can be part of or combined with potential data infrastructure for AI/ML observability and automation.

4.3.5 State-optimized and data-mesh-ready RAN design

Description and motivation

Design of 6G RAN functions should leverage the opportunities provided by state-of-the-art data mesh and state management systems. 6G RAN will increasingly become data-driven, e.g., by natively incorporating AI/ML algorithms. Unlike traditional data management infrastructures, the distributed nature of a data mesh architecture can inherently cope with such ubiquity of data production and consumption in 6G RAN. Similarly, a state management system would be able to maintain state information of RAN entities in a consistent and scalable manner [11].

Examples illustrating the principle

a) State management: Storing and maintaining selected state information parameters could be handled by a shared layer/subsystem that grants the defined read/write rights to the individual RAN functions. Thus, RAN functions would be relieved from the burden of storing and maintaining state information. In addition, this would also remove undesirable redundancy since the same state information is not maintained by several NFs in parallel. Besides these potential benefits, downsides have to be understood properly as well. Efficient state management is extremely crucial for the overall performance of many RAN functions. In such cases, implementation-specific solutions to state management may be the superior solution.

b) A data mesh can support distributed and, if applicable, domain-specific data producing and consuming RAN functions. Targeting an infrastructure to efficiently handle huge amounts of data, setting up data pipelines between producing and consuming NFs in a scalable manner. This could enable ML-driven features for the 6G RAN.

Impact to functional architecture (future standard)

When specifying the functional architecture of the 6G RAN, it is important to understand how 6G RAN can benefit from features of state-of-the-art data mesh and state management architectures. While many of such features would probably rather impact implementation and deployment aspects of the 6G RAN and hence not have any impact on the standardization of the RAN functional architecture, some features may have indirect consequences on the RAN functional design. Potential examples include cloud-friendly data models and protocols, support for state management principles such as observability, scalable transaction management (to handle race conditions), and analytics.

5 Conclusion

This research report has identified and analysed different architecture principles relevant for a cloud-friendly future 6G RAN architecture. As part of the creation and reviewing of this report it has been noted that many of the identified architecture principles for cloud-based deployment of 6G RAN are also applicable to non-cloud deployment.

As stated earlier, the report should be considered as early research for 6G. It is the hope of the authors that the consideration of these principles will be useful input to the standardization of the 6G RAN functional architecture, protocols and interfaces in 3GPP and O-RAN, as well as triggering further research into more detailed impacts and requirements on future standardization, implementation and deployment architectures.

Finally, the authors like to thank all the reviewers (O-RAN and company internal) for all the feedback to the research report.

References

[1] Hexa-X, “Deliverable D1.2 Expanded 6G vision, use cases and societal values”, Available: https://hexa-x.eu/wp-content/uploads/2022/04/Hexa-X_D1.2_Edited.pdf

[2] 3GPP, “3GPP TR 23.700-18, Study on system enabler for service function chaining. V18.0.0”, Available: https://www.3gpp.org/ftp/Specs/archive/23_series/23.700-18/23700-18-i00.zip

[3] ITU-T Focus Group on Disaster Relief Systems, Network, “Overview of Disaster Relief Systems, Network Resilience and Recovery”, Available: https://www.itu.int/en/ITU-T/focusgroups/drnrr/Documents/fg-drnrr-tech-rep-2014-1-Overview.pdf

[4] 3GPP. “3GPP TS 23.501, System architecture for the 5G System (5GS), v17.6.0”, Available: https://www.3gpp.org/ftp/Specs/archive/23_series/23.501/23501-h60.zip

[5] 3GPP, “3GPP TS 38.413, NG-RAN; NG Application Protocol (NGAP), v17.6.0”, Available: https://www.3gpp.org/ftp/Specs/archive/38_series/38.413/38413-h60.zip

[6] 3GPP, “3GPP TS 37.483, E1 Application Protocol (E1AP), v17.6.0”, Available: https://www.3gpp.org/ftp/Specs/archive/37_series/37.483/37483-h60.zip

[7] IETF, “RFC 9260 Stream Control Transmission Protocol”, Available: https://datatracker.ietf.org/doc/rfc9260/

[8] Mwanje, S. et al., “Cognitive Autonomy for Network Optimization”, in: Mwanje, S. et al., “Towards Cognitive Autonomous Networks: Network Management Automation for 5G and Beyond”, pp. 301-343, Wiley, 2020.

[9] Rose, S. et al, “Zero Trust Architecture”, Special Publication, National Institute of Standards and Technology, Gaithersburg, MD, Available: https://doi.org/10.6028/NIST.SP.800-207

[10] CNCF, “Cloud Native Definition 1.0”, Available: https://github.com/cncf/toc/blob/main/DEFINITION.md

[11] Google Inc., “Architecture and functions in a data mesh”, Available: https://cloud.google.com/architecture/data-mesh/