D5.4

SECOND REPORT ON 5G NETWORK ARCHITECTURE OPTIONS AND ASSESSMENTS

The 5G-SMART project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement no 857008.

Second report
on 5G network architecture options and assessments

Grant agreement number:	857008
Project title:	5G Smart Manufacturing
Project acronym:	5G-SMART
Project website:	www.5gsmart.eu
Programme:	H2020-ICT-2018-3
Deliverable type:	Public
Deliverable reference number:	D21
Contributing workpackages:	WP5
Dissemination level:	Public
Due date:	30th November 2021
Actual submission date:	30th November 2021
Responsible organization:	Orange
Editor(s):	Dhruvin Patel, G. Madhusudan
Version number:	V1.0
Status:	Final
Short abstract:	This deliverable presents 5G network architecture concepts from device to cloud aspects, addressing the requirements of the 5G-SMART's smart manufacturing use cases. It also describes the network architecture assessments of the selected aspects.
Keywords:	5G network architecture, Edge cloud, TSN, Device architecture, NPN operation model

Contributor(s):	Bipin Balakrishnan (Ericsson) Dhruvin Patel (Ericsson) Finn Pedersen (Ericsson) G. Madhusudan (Orange) Janos Harmatos (Ericsson) Joachim Sachs (Ericsson) Marilet De Andrade Jardim (Ericsson) Markosz Maliosz (BME) Ahmad Rostami (Ericsson) Matthias Wosnitza (Ublox) Mohammed Zourob (Ericsson) Sylvia Lu (Ublox)

Disclaimer

Executive summary

5G Non-Public Networks (NPN) play a key role in enabling 5G communication services for advanced smart manufacturing applications. These 5G NPNs consist of various components, and, when combined and configured in a proper fashion, can support demanding smart manufacturing application requirements. To ensure end-to-end (E2E) resiliency and deterministic communication performance, there must be a systematic investigation of each component involved in 5G NPN deployments. Following the first 5G-SMART deliverable D5.2 on network architecture options and its investigation, this report takes a further leap by elaborating on the architecture concepts related to devices, Edge cloud, Quality of Service (QoS), and resiliency. Given the importance of 5G System integration with Ethernet-based networks and Time-Sensitive Networking (TSN), a simplified device architecture is proposed that provides support for Device Side-TSN Translator (DS-TT) in a 5G network. Edge cloud service models defined in the first deliverable are taken as baseline for the extensive analysis when integrated with different NPN deployment models, namely Standalone NPN (SNPN) and Public Network Integrated NPN (PNI-NPN). In addition to this, the report extends 5G-TSN integration aspects with Edge computing real-time capabilities. A new TSN interworking function is proposed which enables TSN Frame Replication and Elimination (FRER) function in a virtualized environment. The investigation addresses the open challenges on how high-reliability functionality (e.g., FRER) can be realized in an integrated 5G Edge computing model. A high-level network reliability analysis with focus on physical and virtual components involved within NPN deployment model is performed.

5G NPNs enable novel operation models in smart manufacturing eco-system, here the roles and responsibilities for setting up and operating the 5G network can be distributed among several stakeholders, i.e., the public mobile network operators (MNOs), the industrial parties who use the 5G NPN services and 3rd parties. This results in many theoretically feasible operation models for a 5G NPN, each with its own advantages and disadvantages. The report provides an investigation that results in nine plausible operation models considering today's practical considerations. Additionally, we define a framework to qualitatively analyze the operation models and use it to evaluate and compare the identified operation models.

1 Introduction

Industry 4.0 use cases are usually demanding in terms of End-to-end (E2E) Quality of Service (QoS) – low latency combined with high reliability and availability. This leads to the consideration of various architectural options for the mobile communication technology that are different from conventional mobile broadband services. It is primordial to keep in mind that meeting the objectives of smart manufacturing applications involves an overall system perspective and, hence, includes in addition to the 5G system (5GS), both end system of the communication, i.e., the devices and the Edge cloud hosting smart manufacturing applications. This document aims at addressing system aspects from device to cloud, providing information on how the 5G system is integrated with Edge computing and can ensure E2E deterministic, reliable communication services. This document addresses the topic of resiliency from the perspective of both the 5G system and the Edge cloud architecture. Resiliency is a broad term that encompasses, among other disciplines, redundancy, fault tolerance, and restoration. An in-depth study of redundancy in the 5G system and the Edge cloud is performed in this document.

The NPN deployment options contribute to challenges that are specific to given options. These are also examined in this document, particularly in the context of the Edge cloud architecture and deployment. Going beyond NPN deployment models, the deliverable proposes NPN operation models that specify the assignment of roles to the stakeholders. This helps clarify the roles that can be taken by different stakeholders such as mobile network operators (MNOs), 3rd parties and the industrial parties in deploying, managing, and operating communication infrastructures for smart manufacturing applications.

1.1 Objective of the document

The goal of this document is to propose advanced network concepts and investigate various aspects of the NPN deployment options. In particular, the document highlights open issues and investigates different NPN options for:

- Evaluation of the operation models in the context of different NPNs deployment options,
- Integration of the Edge cloud with 5G NPNs with different deployment models,
- Integration of the Edge cloud with Time-Sensitive Networking (TSN)-based networks,
- High-level reliability analysis.

The focus will be on 5G Stand Alone (SA) architecture. 5G Non-Stand Alone (NSA) is not considered in some contexts. 4G networks and other Internet of Things (IoT) variants such as LTE-M[1] and Narrowband IoT (NB-IoT) are out of the scope of this document.

[1] https://www.gsma.com/iot/wp-content/uploads/2019/08/201906-GSMA-LTE-M-Deployment-Guide-v3.pdf

1.2 Relation to other work in 5G-SMART

Work for this deliverable takes input from 5G-SMART deliverables D1.1 [5GS20-D11] and D5.2 [5GS20-D52], where 5G-SMART use cases are defined, and the initial network architecture investigation is described. D5.2 [5GS20-D52] provides details on the architectural models of the 5G network from a deployment and operation point of view. In particular, the report proposes NPN operations models by defining roles and stakeholders in a smart manufacturing ecosystem. D5.2 brings together all the relevant 5G technical enablers including integration of 5G system with TSN, 5GS support for LAN type services, network slicing and Edge computing. The report provides early systematic deployment validation analysis when such technical enablers are considered for different deployment options. It also proposes 5G-SMART use case relevant edge cloud service models and use case relevant Edge cloud service models. 5G-SMART's deliverable D1.3 [5GS21-D13] investigates the relationship that can be built between different stakeholders in the smart manufacturing eco-system. The work from D1.3 is taken into consideration for the investigation of the operation models in this document.

1.3 Structure of the document

The focus in D5.2 was principally on the 5GS, and the device and Edge aspects were touched upon but not developed in detail. The current report builds upon D5.2, targeting to complete the investigation of network architecture aspects. Figure 1**Error! Not a valid bookmark self-reference.** shows the overall structure of the document showing methodology chosen to achieve the expected results. The deliverable is the elaboration on the new architecture concepts covering E2E aspect of the network architecture, from device to cloud. Further, we define a framework to qualitatively analyze the NPN operation models and use it to evaluate and compare the identified operation models. The report also examines in Section 2.1 the device architecture needed to support a variety of smart manufacturing use cases. This includes integration with TSN and type of 5G communication service supported by the end device. Resilience, which includes redundancy, is a major topic in this report. An overview of various resilience mechanisms available in the 5G standards is provided in Section 2.2

In Section 2.3 the report performs a comprehensive analysis of the integration of the Edge cloud in the context of different NPN deployment options. It also does a deep dive into the Edge integration with TSN (Section 2.4), focusing on adding Edge support for the TSN frame replication functionality (FRER). A high-level reliability analysis is performed in Section 3.2. The Edge sections also develop this theme of resilience in the context of Kubernetes clusters and high availability of FRER mechanisms in the Edge cloud. Building upon the introduced notion of an NPN operation model in D5.2, this report develops this topic in depth in Section 4.1 and provides a qualitative analysis of some carefully chosen operation models.

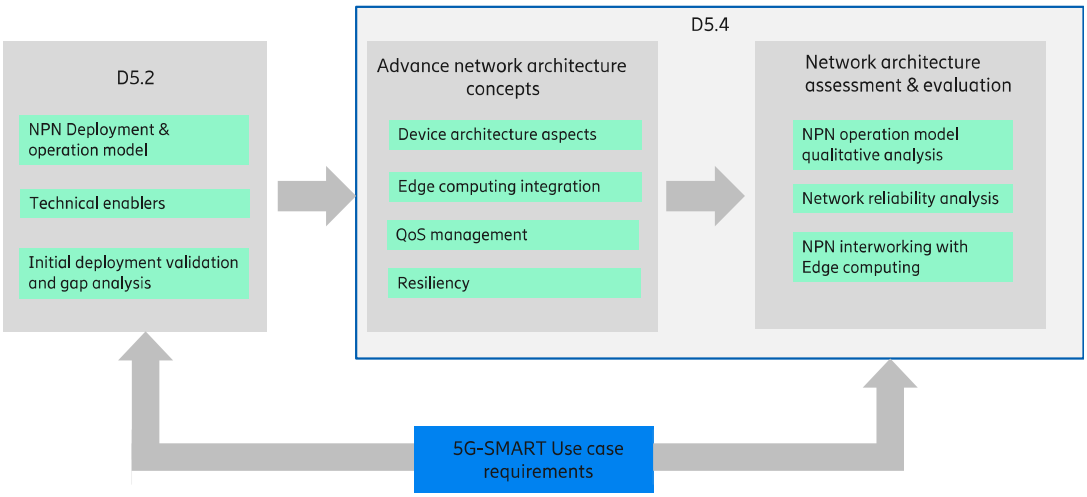


Figure 1 Structure of the document and relation to D5.2 and use cases

2 Advanced network architecture concepts

This section provides details on the investigated advanced network architecture concepts. These concepts include device architecture, resilience aspects of 5G and Edge cloud integration with NPN and TSN. The section completes the deep dives of network architectures investigation which are important to consider when deploying 5G network for smart manufacturing.

2.1 Device architecture concepts

2.1.1 Functional device architecture overview

Given the wide range of 5G industrial applications, diverse communications capabilities would be required for their respective industrial devices. Due to this diversity, communication module and communications technologies used between the different building blocks of a device will have to fulfil the respective demands of the application using the device. It is safe to assume that a single implementation will not fulfil the complete variety of communications characteristics for all applications in an optimized way, considering aspects like power consumption, size, and complexity. On the other hand, implementing special modules for each characteristics profile causes market fragmentation that could prevent the positive effects from economies of scale. Hence, through analyzing the demands of various use cases and respective devices, one can group them into a limited set of devices (hereafter mentioned as device themes) based on their communication characteristics. Such industrial 5G device's themes would lay the groundwork for a thorough device classification in the future, and would serve the purpose of:

- Giving guidance to an application designer to choose the right device communication capabilities based on the needed traffic patterns and characteristics
- Identifying the collection of features supported by the respective device
- Being a foundation for defining test cases within a specific class in order to ensure specific industrial 5G device class will meet its required E2E performance.

The 3rd Generation Partnership Project (3GPP) standard includes support for different communication needs, in order to meet defined communication requirements in certain deployments/scenarios and, of course, to allow efficient and good implementations. This allows a more tailored approach and relative optimizations for a device targeting a specific theme only compared to a device that would comply to all existing theme e.g., considering power consumption, complexity and size. So, a device theme with defined communication characteristics together with standardization efforts could be enablers for certain implementation optimizations, e.g., targeting power consumption size and complexity reductions. However, specific implementation features, and functions are not used to define a theme.

Through the demands of various use cases, three major communication characteristics topics could be identified. The first theme, massive industrial internet of things (IIoT), exhibits communication characteristics such as energy efficient (battery-driven) communications paired with low throughput (up to a few Mbit/s as, e.g., required for industrial wireless sensors in [3GPP21-22104]), low active duty and long inactivity periods, no essential time sensitive data deliveries, and the communication that can tolerate temporary data loss. Devices belonging to this theme can generally have implementation optimizations that allow for low power consumption, small form factor and potentially low cost. The second theme, broadband IIoT, is about very high throughput (high bandwidth), limited latency and defined reliability. Finally, the third theme, time-critical IIoT,

assumes traffic-related characteristics such as ultra-low latency, ultra-high reliability, supporting the highest demands for time sensitive applications. This theme also includes Ethernet-based time-critical communication and in particular TSN.

Communication modules and intra-device communication technologies are expected to be designed and optimized for a specific theme, to satisfy the expected characteristics of the theme. Those optimizations are needed as these themes exhibit contradicting characteristics that cannot be simultaneously and optimally addressed by a single module. Ultimately, however, whether a communication module and the respective building blocks of a device are optimized to cover one specific theme, or if they cover some aspects of other themes to a certain extent is a product design decision.

Further Industrial 5G Device Considerations

In addition to the above defined themes, additional 5G communication features or attributes are important to further specify the capabilities of industrial 5G devices. Such attributes are, e.g., supported frequency bands, number of antennas, etc. The attributes are not modifying the key characteristics of the above themes.

Apart from communication themes, and attributes, industrial 5G devices can be either embedded devices or gateway devices. A gateway device could serve multiple external industrial end devices (e.g., sensors and actuators) and applications within a local area. Hence, for full flexibility, it supports the "aggregated" traffic needs for connected applications/end devices i.e., all earlier mentioned themes (massive IIoT, broadband IIoT and time critical IIoT) but it might also be limited e.g. to the broadband IIoT theme which then also could support the massive IIoT theme.

Prior to deployment, devices belonging to the different themes are expected to undergo various levels of testing/certification/screening based on their reliability, latency thresholds, power consumption, environmental conditions, etc.

Multi Theme Devices and Support

A gateway is a special category of device that supports traffic from potentially multiple end devices. One may distinguish between gateways that are dedicated to a certain type of traffic and for a specific application/use case and those that are general purpose in nature and support many different types of traffic flows. In the first category would be gateways that are meant for time critical applications which have strong demands on the performance of the gateway so that timing constraints and very low latency are respected. On the other hand, gateways handling broadband IIoT could also support the massive IIoT theme but not benefit from all potential optimizations (e.g., for power consumption reduction) that is possible with only massive IIoT.

Device Authentication

Authentication of the devices in an industrial environment is critical and technology employed to do so can be realized with two options. The first authentication option would be the 3GPP-based authentication scheme such as for 5G. The second authentication option would be non-3GPP-based authentication schemes, such as those schemes based on Extensible Authentication Protocol (EAP) [IETF-EAP78], which is widely used in the IEEE 802.11 standards.

Device aspect of Ethernet bridging details

A wide range of the industrial applications require industrial end devices to interwork with existing Ethernet-based industrial networks. In this regard, 3GPP has defined a special entity called Device Side TSN Translator (DS-TT) for 5G networks. The DS-TT is one of key components specified by the 3GPP to ensure integration with Ethernet-based industrial networks (including TSN). The DS-TT connects to the UE protocol stack, as shown in Figure 2. Also, figure depicts the control plane communication towards the 5G core network via non-access stratum (NAS) data exchange. For user plane communication, the DS-TT communicates through the UE via an Ethernet Packet Data Unit (PDU) session towards User Plane Function (UPF) and the Network TSN Translator (NW-TT), which connects on the other side of the 5G system to an Ethernet network. The user plane communication can also go from one device DS-TT directly to the UE and DS-TT of another device. In this case the Ethernet PDU session from the first device is bridged in the UPF to another Ethernet PDU session that connects the UPF to the other device. For the external Ethernet-based industrial networks, the DS-TT interface is seen as Ethernet compatible port of the 5G network, also referred to as 5G bridge in this context.

Another important functionality for the DS-TT is to ensure proper functioning of the time synchronization operations. Time synchronization can be directed to the device via the 5G networks, or it can originate at a grandmaster clock connected to the device for (uplink) time synchronization through the 5G network. In the former case, the device acts as the egress side of the 5G system for the handling of PTP messages, and in the latter the device is at the ingress side of the 5G system. For egress-side PTP handling, the DS-TT has to perform the following functions: timestamping of PTP messages, calculation of a residence time, modifying the PTP message fields (setting the correction field), re-generating the PTP Announce message. For ingress-side PTP handling, the DS-TT has to perform timestamping, modify the PTP message by adding the timestamp, and forward the message towards the UPF / NW-TT. Additionally, depending upon the scenario where 5GS acts as the GM, DS-TT may implement the Grand Master clock functionality.

At last, according to 3GPP standard the DS-TT should collect the UE-DS-TT residence time (used to calculate the 5G bridge delay), propagation delay, topology information and other IEEE specified information and report this information within a port management information container (PMIC) via control signaling to the Time Sensitive Network – Application Function (TSN-AF) in the 5G core network. The TSN-AF sends configuration parameters to the DS-TT via PMIC, and the DS-TT should be able to set the configuration as provided.

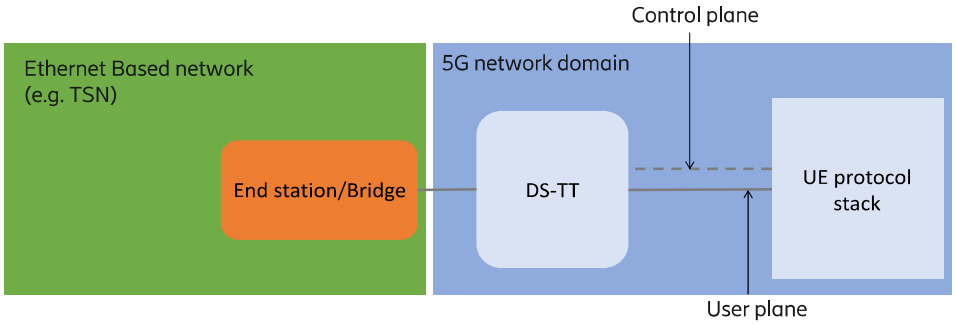


Figure 2 DS-TT view for integration of industrial networks with 5G

2.1.2 5G device architecture for the 5G-integrated Ethernet-based TSN network

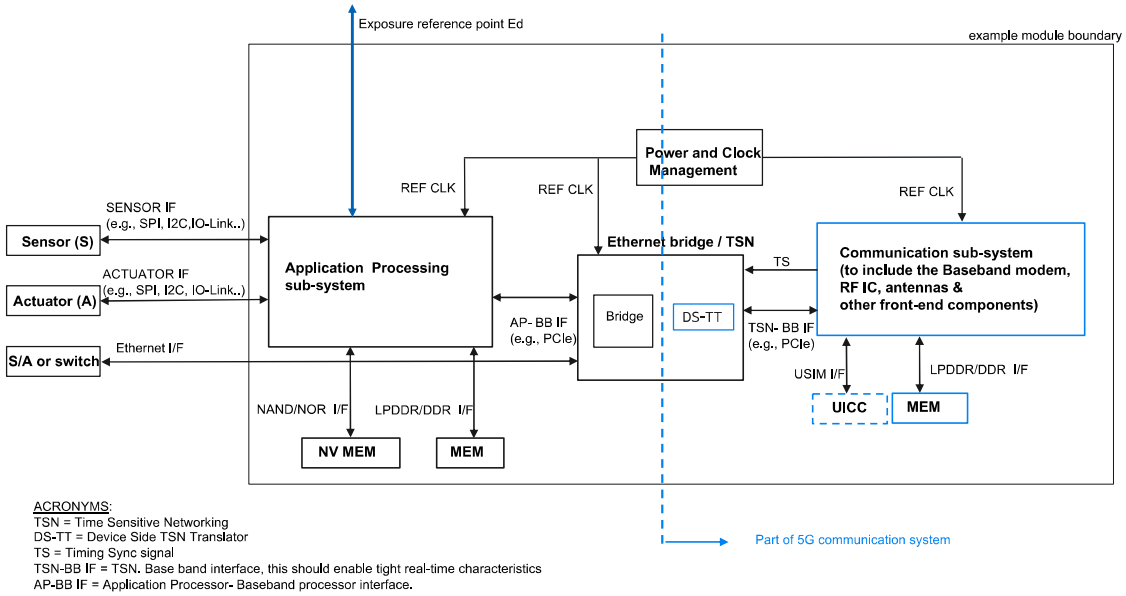


Figure 3 depicts a generic architecture of an industrial device that also has TSN capability. Here TSN function is shown as standalone block for emphasis but could be integrated for example to the communication sub-system. Figure 3 also shows typical building blocks (i.e. UICC, MEM, AP-BB-IF, TSN-B-IF) of embedded end device that can also see in typical mobile platforms. The details around this can also be found in reference[2]. Furthermore, the report highlights new enhancements which are tailored for industrial use cases, for example integration with TSN network. Such an architecture would correspond to a time critical IIoT theme device in the Section 2.1.1, The industrial device could be made available as a module (rectangle marked by “typical module boundary” in figure 3). The key functions in the module are presented and their mapping to possible physical implementations (e.g., interfaces) are shown where it is deemed more relevant. The functional components can be split broadly into two groups - those that are part of the 5G communication system (represented in blue coloured boxes) and those that relate to running the application or interfacing to sensors/actuators (represented in black coloured boxes). For the sake of simplicity, the DC power source (e.g., battery) and clock source is shown as common within the Power and Clock management module and as owned by the application domain.

[2] https://www.mipi.org/about-us

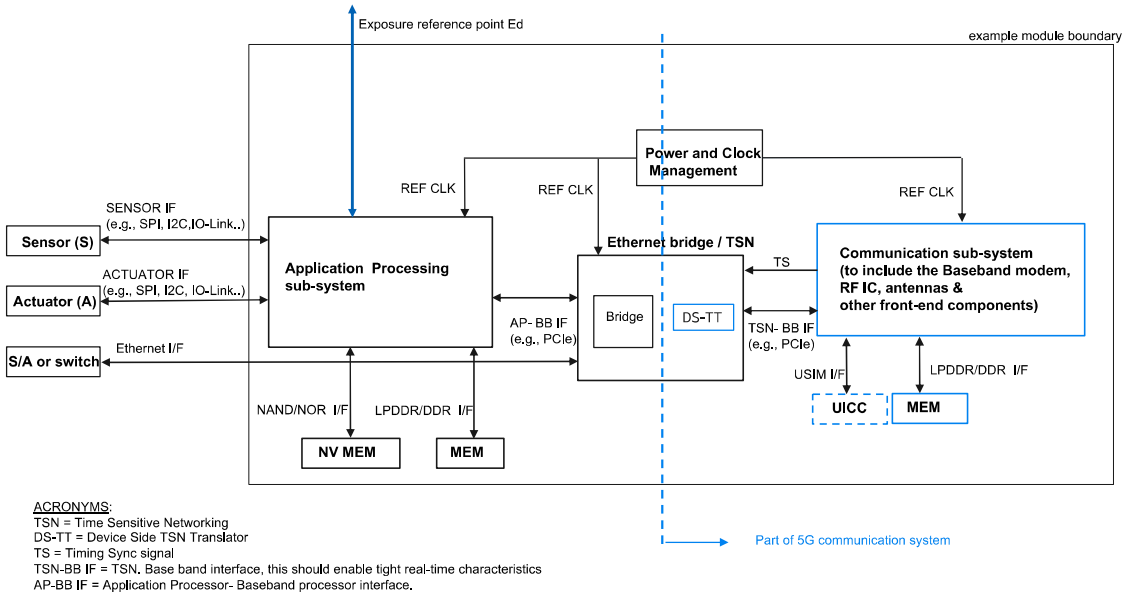


Figure 3: Industrial device generic architecture (with TSN capability).

The core functions in the 5G communication sub-system are the Radio Frequency (RF) front end functions such as filtering etc., RF processing function and baseband processing function (along with its memory). These form the core part of the communication sub-system. The authentication function could be with Universal Integrated Circuit Card (UICC) that hosts the USIM application and is shown optional, given the industrial device might go with non-3GPP based authentication methods (such as those based on Extensible Authentication Protocol (EAP)). Another, physically non-removal implementation mapping for authentication function is to have an embedded UICC (eUICC), which is not shown in the Figure 3. In case of eUICC, there is no need to change the SIM card and instead this programmable sim card is embedded within a device during manufacturing. Then the eUICC can be provisioned remotely and enables to change operators without having to physically change the UICC card.

One notable component in the 5G communication sub-system is the DS-TT, which is boundary between the ICT (5G) system and the OT system. DS-TT has further sub-components (e.g., a generalized PTP (gPTP) instance), which are not detailed in the figure, that need strict timing reference signals. Hence the Timing Sync (TS) signal between the Baseband processing function and DS-TT is shown explicitly. This TS signal could be implemented with an IO signal when the DS-TT block is implemented on a separate chip other than the baseband. Note that for accuracy reasons, it should be driven from hardware state machines (as against the software driven IO signal) to keep its error within a very small fraction of an assigned total 5G system timing accuracy budget. Another notable interface is the data interface between the DS-TT sub-module and the Baseband processor, which should be a high-speed interface without any software (e.g., PCI express) involved to move the data. For devices without time critical functionality (such as those falling into massive IIoT and broadband IIoT themes in Section 2.1), slower interfaces can be considered, including even those interfaces (e.g., USB) with software programmed data transfers, if it can satisfy the application requirements.

Device themes and their influence on application processing sub-system

The different device themes will have different implementation needs and could influence the application processing sub-system. A typical application sub-system mainly consists of the CPU system running the application and needed memories. This sub-system interfaces to the sensors/actuators or possibly even an Ethernet switch, behind which the OT field buses could be residing.

The massive IIoT themed devices may only have a low-end CPU/MCU or even use an additional CPU in the base-band processing function and may not have the Ethernet interface. In addition, this theme may use a 5G communication sub-system tailored for low complexity and low power consumption.

The broadband IIoT and time critical IIoT themed devices could have more functionality in the application sub-system for example to cater to specialized video processing using GPUs or image processing functional units. More advanced application sub-systems could have dedicated hardware accelerators for sensor fusion processing etc. The industrial device may be configured via the OPC-UA interface, which is then most probably connected to the application processor.

2.2 End-to-End resilience over a 5G network

Industrial IoT applications often demand a high level of availability. Depending on the use cases it may be 99.9999% ("six nines") or even higher. Six nines correspond to an unavailability of 31 seconds in a year. It should be noted that this is E2E and hence has an impact on the entire chain of links and nodes between the UE (end device) to an application entity. 3GPP has raised the bar for 5G RAN URLLC availability to six nines. Traditionally the approach to improve availability has been to provide increased levels of redundancy. However, redundancy is just one aspect of a broader field called resiliency.

Resilience is the ability of the network to provide and maintain an acceptable level of service in the face of various faults and challenges to normal operation [ResiliNets].

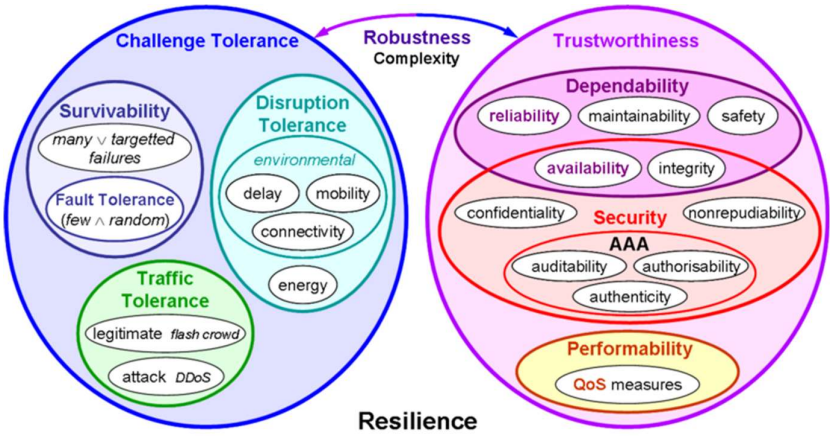


Figure 4 Different disciplines involved in ResiliNets models[3]

[3] www.resilinets.org

Figure 4 gives an overview of the different disciplines involved in the ResiliNets model. As can be seen, many disciplines and domains are involved under the umbrella of resilience. We will focus on a few areas that are important from the architecture and deployment point of view for smart manufacturing applications. The following model of enablers is proposed to ensure robustness and resilience [ResiliNets].

Table 1: ResiliNets enablers

Enabler	Related Disciplines	Definition, examples
Security and Self-Protection	Security and dependability	Self-protection is implemented by a number of mechanisms, including but not limited to mutual suspicion (authentication, authorization, accounting) and additional conventional security mechanisms (confidentiality, integrity, nonrepudiation)
Connectivity and Association	Disruption tolerance, Survivability	Connectivity and association among communicating entities should be maintained, when possible, but information flow should still take place even when a stable end-to-end path does not exit
Redundancy	Dependability, Survivability, Performability	Redundancy refers to the replication of entities in the network, generally to provide fault-tolerance. In the case that a fault disables part of a system, the redundant parts are able to operate and prevent a service failure. Redundancy can be further categorized in two ways: - degree (*k*-redundant) - type (hot spare, active load balance, on-demand) Redundancy types: - spatial redundancy (e.g., additional nodes and links) - temporal redundancy (e.g., redundant protocol information) - information redundancy (e.g., supplementary storages)
Diversity	Disruption tolerance, Traffic tolerance, Performability	Diversity consists of providing different alternatives so that even when **challenges** impact some particular alternatives, other alternatives prevent degradation from **normal operations**. The degree of diversity is the number of different alternatives. Diverse alternatives can either be simultaneously operational, in which case they **defend** against **challenges**, or they may be available for use as needed to **remediate** - spatial diversity (e.g., links of different nature, heterogeneous nodes) - temporal diversity (e.g., various timer values) - operational diversity (e.g., implementations)
Multilevel Resilience	Dependability, Survivability	An overall resilient system requires resilience at the various internal levels of its implementation. In the case of the global network, multilevel resilience is needed along three orthogonal dimensions: data, control and management planes

Context Awareness		Context awareness is needed for resilient nodes to monitor the network environment (channel conditions, link state, operational state of network components, etc.) and ***detect* adverse events or conditions**.

The following subsections examine some mechanisms provided by 5G that contribute to the enablers listed above. The goal is not to be exhaustive but to highlight some features that are important for smart manufacturing scenarios and have an impact on the 5G architecture and deployment. We also focus on network-level features rather than application-level ones such as FRER for TSN. While this section focuses on the 5G concepts and enablers, section 2.2.1will provide an analysis for smart manufacturing scenarios with emphasis on redundancy and availability/reliability.

2.2.1 Redundancy

The 5G system offers several mechanisms to support redundancy. In this section, mechanisms defined in the 3GPP specification TS 23.501 [3GPP20-23501] are described.

Dual Connectivity

One of the major resiliency improvements with 5G is the ability to combine access and core mechanisms to achieve end-to-end user plane redundancy. This is referred to as Dual Connectivity based end to end Redundant User Plane Paths [3GPP20-23501]. The duplicated traffic originating from the same application are associated to two redundant PDU sessions based on the User Equipment (UE) using Route Selection Policy or UE local configuration. One PDU Session is established from the UE via gNodeB 1to UPF 1 (see Figure 5) acting as the PDU Session Anchor (PSA), and the other PDU Session from the UE via gNodeB 2 to UPF 2 also acting as the PSA. The redundant user plane set up applies to both IP and Ethernet PDU Sessions.

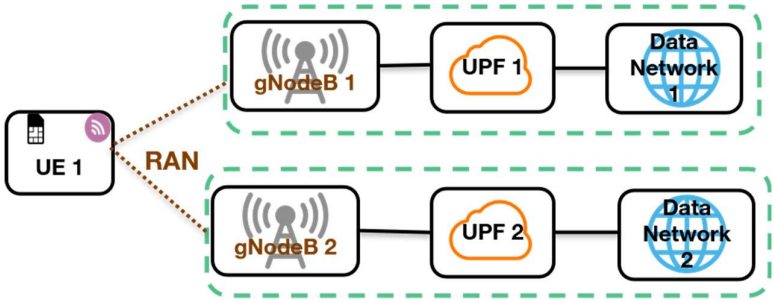


Figure 5 Dual Connectivity

From the architecture and deployment perspective it is important to ensure that the above two paths are disjoint:

- All UEs using critical communication support dual connectivity
- There is sufficient RAN coverage for dual connectivity in the target area (factory floor, etc.)
- UPF deployment and transport between RAN and UPF supports redundant user plane paths
- The operation of the redundant user plane paths is made sufficiently independent, to the extent needed by the smart manufacturing applications needs, e.g., independent power supplies.

Dual connectivity is necessary for very high reliability requirements for both S-NPN and PNI-NPN deployments. It should be noted that this redundancy applies only to the user plane and not to the control plane.

Redundant user plane paths based on multiple UEs per device

Another approach to achieving redundancy is by the device having multiple UEs. The RAN deployment ensures redundant coverage with several gNodeBs in the deployment area. The network operator/integrator ensures proper configuration so that each UE uses a different gNodeB in order to provide disjoint paths. The network operator can further ensure that the transport and core networks offer disjoint paths to the Data Network (DN). Figure 7 illustrates the scenario for a device with dual UEs with disjoint paths to the DN.

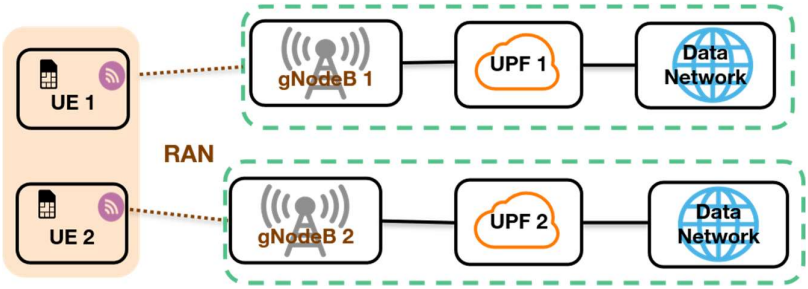


Figure 6 Redundancy based on multiple UEs in the device

Redundant transmission at transport layer

Application-level mechanisms exist for packet duplication in order to decrease the loss probability of packets. An example is the FRER mechanism provided by TSN. The 5G system allows for packet duplication and elimination of redundant packets at the transport layer i.e., the backhaul between the 5G RAN and the UPF. Only one tunnel connecting the RAN and the UPF is needed for this mechanism. For downlink transmissions the UPF takes care of duplicating the data and the 5G RAN eliminates the received duplicated downlink data. The reverse procedure is applied for uplink transmissions i.e., 5G RAN duplicates the uplink data and the UPF eliminates the duplicates.

Redundant transmission on N3 interfaces

This is similar to the redundant transmission at the transport layer albeit with several important differences. N3 is the interface between the gNodeB and the UPF acting as PDU Session Anchor (PSA),

- Two independent N3 tunnels are setup instead of just one.
- To ensure the two N3 tunnels are transferred via disjoint transport layer paths, the control layer should provide different routing information in the tunnel information (e.g., different IP addresses or different Network Instances), and this routing information should be mapped to disjoint transport layer paths according to network deployment configuration.

Summary

We have described several redundancy mechanisms offered by the 5G system. These mechanisms are not mutually exclusive and may be combined. For instance, redundant transmission on N3 may be combined with dual connectivity.

All the mechanisms are dependent on the deployment strategies of the network operator of the NPN based on the requirements provided by the user and involve the integrator of the NPN in making sure that the different network elements function properly together.

Some mechanisms are more appropriate for certain 5G NPN deployment models than others. NPN deployments have been described in detail in the previous architecture deliverable of 5G-SMART D5.2 [5GS20-D52]. Table 2 provides a short summary.

Table 2 NPN deployment models

Deployment Option no.	NPN deployment options	Characteristic/details	3GPP terminology
NPN 1	Standalone NPN	All NPN functionalities are on-premises. NPN is a fully separate physical network from the Public Network (PN) with dedicated NPN ID. However, dual subscription with NPN and PLMN is possible. Access to PLMN services can be realized via an optional firewall connection and roaming agreement.	SNPN
NPN 2	Shared RAN	NPN is based on 3GPP technology with its own NPN ID. Only the RAN is shared with the PLMN, all other network functions remain segregated, also data flows remain local. It can be realized by: • Multi-Operator Core Network (MOCN), where two or more entities are sharing eNodeB/ gNodeB and spectrum • Multi-Operator RAN (MORAN), where two or more entities are sharing gNodeB with non-shared spectrum	
NPN 3	Shared RAN and control plane	NPN is based on 3GPP technology and RAN shared with the PLMN. The network control plane is hosted by the PLMN. Data flows remains local.	PNI-NPN
NPN 4	NPN hosted by the PN	NPN traffic is off premise but treated differently through Network Slice instances and dedicated DNNs.	

Redundant transmission at the N3 interface is more useful for NPN deployment models NPN3 and NPN4 (PNI-NPNs) as the two ends of N3 (NG-RAN and UPF) are not both on the factory floor. On the other hand, a standalone NPN model that chooses to collocate the UPF with the gNodeB does not need redundant transmission on the N3 interface.

Regardless of the NPN deployment model, very high availability services need to ensure that the UE has connectivity to at least two gNodeBs in order to provide the level of reliability that is required.

Section 3.2will provide an analysis of the reliability requirements and its impact of the deployment choices. The above mechanism enables high availability and reliability for user plane transmission within 5GS and not for the control plane transmission between UE and 5GS.

2.2.2 Support for network reliability with Network Functions (NF) sets and stateless functions

As compared to older 3GPP cellular communication generations, the 5G Core system moves from a fixed set of often hardware-based functions to a software based and cloud friendly architecture. A network function is an element of the network providing a well-defined functionality. Interactions between the 5G control plane network function uses a Service Based Architecture (SBA) [3GPP20-23501]. Examples of Network Functions are Access and Mobility Functions (AMF) and Session Management Functions (SMF).

3GPP provides mechanisms to enhance reliability for Network Functions (NF) particularly in the control plane. A NF Set is a group of interchangeable NF instances of the same type, supporting the same services and the same Network Slice(s). The NF instances in the same NF Set may be geographically distributed but have access to the same context data. One way of achieving this sharing of context data is by making the NFs stateless. An NF is “stateless” when the compute and storage resources are separated i.e., the state is stored in an entity like UDSF (Unstructured Data Storage Function). Thus, in the event of a malfunction or failure at the level of one NF instance another instance in the same NF set can take over by retrieving the context information from the UDSF.

Note that a NF can be decomposed into NF services and above-described mechanisms are applicable to NF services via a NF Service Set. The following example is intended to illustrate the importance of this mechanism in the context of smart manufacturing. High E2E reliability, in for instance tight control loops, is primarily a user plane matter. However, there is an interplay between the control plane and user plane. The interface between the SMF and UPF is referred to as N4 in the 3GPP specifications. A regular heartbeat is exchanged between the SMF and UPF on N4. The value of this timer is implementation dependent. When this time has elapsed and the UPF has not received a heartbeat from the SMF, the session context is dropped which means the current PDU session is closed. One way to mitigate this is by having a SMF set and an N4 association between the SMF Set and an UPF. Then any SMF in the SMF Set should be able to manage the N4 association with the UPF and thus keep the user session alive.

A related problem is that, as seen in section 2.2.1 ultra-reliability services will use dual UPFs for user plane redundancy. The SMF NF set will use two different N4 interfaces to these UPFs.

Control plane resiliency mechanisms are of critical importance to NPN3 deployment model as the UPF and the SMF are not in the same administrative domain. Even for NPN1 and NPN2, the SMF may be located in a central room and the two UPFs maybe in different areas of the factory floor (in order to ensure low latency where the factory covers a large area). Consequently, ensuring control plane resiliency is important even in these scenarios.

2.2.3 Restoration mechanisms

Restoration mechanisms are described in 3GPP 23.527 [3GPP21-23527]. These mechanisms have been defined by considering three network areas:

- The N4 interface, between the SMF (Session Management Function) and the UPF (User Plane Function), in other words the link between the control plane and the user plane of the core network
- The User plane itself, at the level of the link between the access and the core or between two nodes of the core network (N3 and N9 interfaces)
- Service Based interfaces

For the first mechanism, procedures are planned in the event of failure or restart of the UPF and SMF functions, and in the event of failure of the link between these two functions. Failure detection is done through "heartbeat" messages broadcast regularly on the N4 interface.

For the second mechanism, hooks are provided in the event of loss of contexts between the access and the core, with procedures to be set up at the access or at the level of a UPF function upon receipt of the indication of or lost context error.

A user plane entity can also detect a link failure on the transmission path through echo messages.

For service-based interfaces (SBI) there are:

- Mechanisms for detecting failure or restarting a NF or a NF service, from the NRF (Network Repository Function), from "heartbeat" messages
- Mechanisms for detecting restart of a producer or consumer of a NF service directly on the signalling links
- Mechanisms for reselecting a NF service instance in the event of failure

2.2.4 Contribution of virtualization and containerization

This subsection does not form part of the standardization of 5G but are enablers to help achieve the resiliency features discussed above. The section 2.4.6 on Edge Computing explores in detail resiliency methods in the context of the Kubernetes platform.

With virtualization, there is more flexibility in redundancy models with several instances simultaneously active in the access network and in the core network. With containers, the cloud-native implementation of 5G functions allows faster instantiation compared to technology based on virtual machines. Beyond redundancy for reliability purposes, instances can be added to improve performance and deal with extra load.

Containers have smaller size, especially compared to virtual machines (VMs), means they can spin up quickly and are better able to support cloud-native applications that can be scaled as needed to

support higher traffic and/or redundancy. Containers carry all their dependencies with them meaning that software can be written once and then run without needing to be re-configured across for instance cloud environments.

Containers enable a micro service architecture, which means that application components can be deployed and scaled more granularly. However, the lack of hardware isolation is the main drawback in the container-based approaches.

2.3 NPN integration with Edge computing

Edge computing provides an ecosystem, where the distributed execution environment (e.g., compute and storage resources) is closer to the location where it is needed in contrast with a remote cloud. The proximity of the edge premise results in reduced latency between a client and the server application, so edge computing can support use cases, where ultra-low latency and high reliability characteristics are crucial. The typical usage of the edge computing in smart manufacturing could be to perform tasks on the edge infrastructure that are intensive in complexity, computation, memory and storage. For example, various analytics and monitoring tasks can be executed on the edge, utilizing the cloud capabilities (e.g., scalability, robustness) and process huge amounts of data locally.

The distributed edge computing with real-time execution capabilities integrated with 5GS and Ethernet-based industrial networks will be able to provide ultra-low end-to-end latency communication service for a wide range of time-critical 5G-SMART use cases. Also, such edge computing solutions enable offloading of time-critical industrial device control applications, such as robot motion control and AGV control.

To sum up, edge computing is not limited only to improve the effectiveness of existing use cases, but it can support new use cases as well:

- Edge can host such resource consuming (industrial control) tasks, which cannot be deployed on a device (due to limited compute resource or battery power). For example, edge computing can enable more complex AI supported control mechanisms for mobile robots (AGVs), process extreme volumes of data and support virtual or augmented reality aided visualization which can improve the efficiency of the service as shown in forward-looking use cases in 5G-SMART Deliverable D1.1 [5GS20-D11] such as Use Case 10: AGV and Real-time Trajectory Adaption with AI for Smart Factories and Use Case 13: AI-assisted Production Quality Management,
- The edge-enabled, centralized device control makes it possible to support extensive coordination between different devices (e.g., AGVs and robotic arms in a collaborative operation) resulting in improvements of the productivity,
- Edge computing enables the balance between local (close to the premise where the data is generated) and central cloud-based data processing.

Considering the end-to-end scope, it is important to investigate how edge computing can interwork with 3GPP NPN deployment options [5GS20-D52]. This section starts with a brief overview of the main edge computing related standardization efforts, and in the rest of the section an in-depth investigation for several edge computing deployment options suitable for both SNPN and PNI-NPN scenarios is provided.

2.3.1 Edge computing standardization overview

Edge computing as seen today has a quite fragmented and evolving ecosystem. Additionally, the standards and business models are in the phase of maturing. The goal of this section is to give a short overview about the status of the main standardization directions.

3GPP

3GPP is a global standardization body for mobile communication technology. Several working groups within 3GPP are focusing on edge computing.

In 3GPP SA2 working group, the technical specification on '5G system enhancements for edge computing' [3GPP21-23548] and '5G system architecture' [3GPP20-23501] specify the details on how user traffic is routed to the appropriate edge application server within 5GS, covering edge application server discovery, UPF selection and connectivity models that enable edge computing for edge-unaware devices. In 3GPP SA6 working group, TS 23.558 [3GPP21-23558] defines an architecture for edge computing support of the devices, which have edge-aware capabilities. The proposed architecture includes an edge enabling layer, which facilitates the communication between the application clients and the servers (e.g., optimized edge application server discovery, network exposure capabilities towards the edge application server). In 3GPP SA5 working group the life cycle management aspect of the application servers in the edge is specified [3GPP21-28814].

ETSI

ETSI Multi-access Edge Computing (MEC) has created an open and standardized IT service environment which allows third-party applications to be hosted at the edge of the mobile network and which is capable of exposing network and context information. It specifies a framework for service delivery, APIs for exposure and programmability, as well as covers management and orchestration. ETSI MEC studies federated architecture to support multi-operator, multi-vendor scenarios. ETSI MEC also makes efforts to propose a synergized architecture leveraging the ETSI MEC and 3GPP specifications [ETSI-MEC20].

GSMA

GSMA specifies an end-to-end high-level architecture and provides edge cloud service description mainly from the telco operator perspective. GSMA also describes stakeholder roles and different business models for telco operators in edge ecosystem [GSMA20-TEC].

5G-ACIA

5G-ACIA investigates how 5G can be used in industrial systems to provide novel capabilities to industrial use cases. As part of this work the role and capabilities of edge computing, its enablers and architecture are analysed, together with how edge computing can be applied to industrial use cases.

2.3.2 Kubernetes basic concepts

Kubernetes[4] is a widely adopted open-source platform for managing containerized workloads and services in a virtualized environment. Kubernetes can also manage virtual machines with the KubeVirt[5] virtual machine management add-on. Kubernetes enables edge computing solutions for a wide range

[4] https://kubernetes.io/docs/home/
[5] https://kubevirt.io/

of deployment models. In this subsection we give a short summary about Kubernetes basic concepts that are referred to later in the document, to help understanding them. It is not intended to give a full and detailed overview about Kubernetes, the interested reader can check the Kubernetes official documentation[6] for deeper insight.

For edge cloud deployments in the data center, typically the hardware part of a data center consists of a network of computing and storage resources including storage systems, servers, networking devices, etc. The software part of a datacenter that manages these resources is the cloud platform, such as OpenStack or Kubernetes.

A *Kubernetes cluster* is a set of *nodes* that can either be virtual machines[7] or physical servers, connected to the same network, to work together to operate as one cloud platform.

Pods[8] are the smallest and most basic deployable objects in Kubernetes. A Pod represents a single instance of a running application in the cluster. Pods contain one or more containers, such as Docker containers. When a Pod runs multiple containers, the containers are managed as a single entity, share the Pod's resources, and will run on the same node. A Pod can be considered as a self-contained, isolated "logical host".

Deployments represent a set of multiple, identical Pods. A Deployment runs multiple replicas of an application and manages the lifecycle of the constituent Pods, such as, automatically replaces any instances that fail or become unresponsive. In this way, Deployments help ensure that one or more instances of applications are available to serve user requests. In addition, a Deployment can also support *horizontal scaling*, i.e., changing the number of Pods within the Deployment.

Services are an abstract way to expose an application running on a set of Pods. A Service gives a single DNS name and IP address for the set of Pods and can load-balance across them.

For our given investigation, Kubernetes is utilized as a reference edge computing platform solution for its integration with NPN deployment options.

5G-SMART Deliverable D5.2 [5GS20-D52] provides a detailed analysis of the three different edge cloud setup models (shown in Figure 7): K3s[9], KubeEdge[10] and Kubernetes Cluster Federation[11].

1. K3s is a lightweight Kubernetes distribution, built for IoT and edge Computing. All components of K3s run on the edge, therefore no cloud-side collaboration is involved (see Figure 7 top part). If K3s is to be used in production environments, there should be a cluster management solution on top of K3s that is responsible for cross-cluster application management, monitoring, etc. (If the on-premise hardware resources enable it, a Kubernetes cluster can also be installed and used as a standalone cluster on the edge.)

[6] https://kubernetes.io/docs/home/
[7] https://www.vmware.com/topics/glossary/content/virtual-machine
[8] https://cloud.google.com/kubernetes-engine/docs/concepts/pod
[9] K3s Lightweight Kubernetes, https://k3s.io/
[10] https://kubeedge.io/en/
[11] Kubernetes Cluster Federation, https://github.com/kubernetes-sigs/kubefed

2. KubeEdge is made to build edge computing solutions to extend the central cloud (Figure 7 middle part). KubeEdge consists of a cloud part and an edge part, both edge and cloud parts are open-sourced.
3. Kubernetes Cluster Federation (KubeFed for short) allows to coordinate the configuration of multiple Kubernetes clusters from a single set of APIs in a hosting cluster (Figure 7 lower part). KubeFed aims to provide mechanisms for expressing which clusters should have their configuration managed and what that configuration should be.

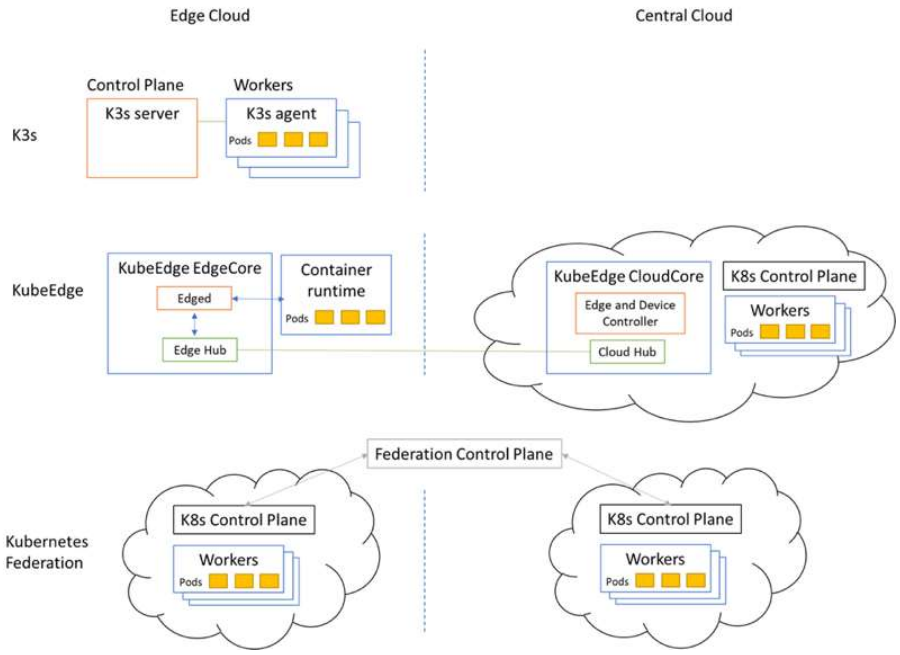


Figure 7 Edge cloud scenarios [5GS20-D52]

In the following subsections we are extending the investigations by integrating edge scenarios with different NPN deployment models summarized in Table 3. In some use cases the edge computing domain will be deployed locally, in the factory premise (due to use case requirements), and SNPN is applied, therefore several combination scenarios of edge computing and SNPN deployment are investigated. Here it is also considered if the infrastructure of the edge and NPN domains are isolated or shared. Since the footprint of the telco operators enables to provide edge computing services for factory enterprises, different scenarios are also investigated where PNI-NPN deployment options are combined with MNO provided edge.

Table 3 Summary of the NPN and edge deployment options

NPN deployment options	Possible edge computing deployment
Standalone NPN	On-premise – Standalone edge
Standalone NPN	On-premise – Federated edge
Standalone NPN	On-premise – Integrated edge and central cloud premises

Shared NPN infrastructure	On-premise edge
PNI-NPN with shared RAN and core control plane	On-premise edge
PNI-NPN hosted by the Public network	Telco/3rd party edge
PNI-NPN hosted by the Public network	On-premise edge

2.3.3 Standalone NPN with private, on-premises edge

As shown in Figure 8, in this scenario all user plane and control plane functions required to operate the NPN are physically located on factory premise. The edge deployment that hosts the latency critical industry applications (e.g., cloud-based mobile robot control) is also on-premises and a dedicated infrastructure is used for the edge domain. Due to the totally separated infrastructure of the edge and SNPN domains, this scenario ensures the highest isolation for the industry applications running at the edge.

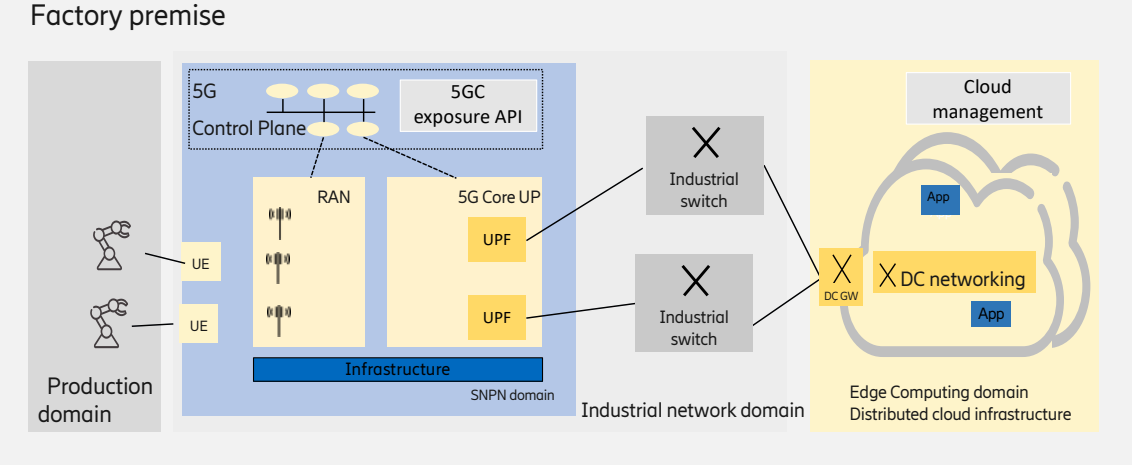


Figure 8 Standalone NPN with private, on-premises edge

The industrial network domain that comprises the SNPN and the wired network segments connects the industrial end devices to the edge computing domain, which can be deployed in different configurations:

- A single, standalone datacenter could be deployed in the factory premise.
- Standalone Edge (standalone) datacenters deployed in different factory buildings realizing a distributed edge infrastructure and each datacentre is managed as separate edge computing cluster. This provides robustness (multiple application instances can be deployed on redundant infrastructure), with high data privacy (sensitive data is kept locally).
- Another operation mode is to manage each datacenter individually as separate Kubernetes clusters, but Kubernetes enables the handling of the multiple clusters in a federated way [5GS20-D52].

- Kubernetes also enables to support such cases, if beside the on-premises edge computing datacenter(s), a central (public or private) cloud premise is also used for hosting the industry applications/application functions with relaxed requirements (e.g., latency). [5GS20-D52].

The private, on-premises edge deployment enables lots of flexibility, however the edge owner[12] is responsible for the handling of the full cloud stack, including:

- The infrastructure layer (e.g., compute and storage resources, as well as cloud networking) has to provide low latency (e.g., near-real time) capabilities. Furthermore, it should enable the redundant deployment of application instances as well as datacenter network level redundancy to ensure high availability.
- The selection and maintenance of the virtualization platform, such as container runtime environment (e.g., Docker[13]) or VM-based ecosystem is needed.
- Maintenance of the orchestration system. The most wide-spread open-source container orchestration platform for containerized workload is Kubernetes, so the proper installation and management of Kubernetes components (e.g., provide the redundancy of the Kubernetes control plane, Kubernetes cluster configuration, etc.) is also required.

Depending on the stakeholder roles, the above tasks should be handled by the industrial party (self-managed deployment), a 3rd party integrator[14] or the MNO. The roles for a selected NPN operation model are defined in section 3.1. In the two latter cases, the industrial party should only perform the application deployment and life-cycle management, while in the first case, the industrial party is responsible for proper operation of the full cloud stack. The on-premises edge scenario enables numerous deployment options, the most specific ones are discussed in the next part of the section.

Standalone edge datacenters

This alternative is suitable for such industry scenarios, where each component of an application software runs locally, at the edge [5GS20-D52]. A typical example could be the edge-enabled control of an AGV, where all components, which are offloaded from the industrial end-device have to be deployed on the on-premises edge.

One deployment option is to have only a single edge datacenter in the factory premise. Even in this case, multiple execution environments can be deployed at the edge in order to support the various requirements of applications, e.g., OpenStack supports virtual machines, bare-metal servers, containers from one control plane or virtual machines and Kubernetes clusters can be run in the same datacentre when the Kubernetes clusters are deployed on virtual machines. Alternatively, multiple standalone datacenters could be deployed, on different hardware infrastructure and connectivity as well as platform capabilities. This enables that the different datacenter infrastructures, as well as platforms, can meet with the requirements (real-time execution environment, support of hardware acceleration, TSN-FRER support, etc.) of the different applications (such as real-time device control applications as well as monitoring, analytics tools). The datacenters are handled as separate

[12] Edge owner is owning the edge infrastructure and includes both hardware and software components
[13] https://www.docker.com/resources/what-container
[14] 3rd party integrator is the stakeholder who provides hardware and software components for deployment and management of the edge and make it ready to use

Kubernetes clusters from the cloud management perspective. Multiple datacenters even enable the support of both containerized and VM-based application deployments.

Federated edge datacenters

Multiple datacenters deployment in the factory can be handled and managed as a federation. This results in high level of reliability (different application instances could be deployed on different datacenter infrastructure), as well as increased scalability can also be provided over the distributed cloud infrastructure as shown in Figure 9.

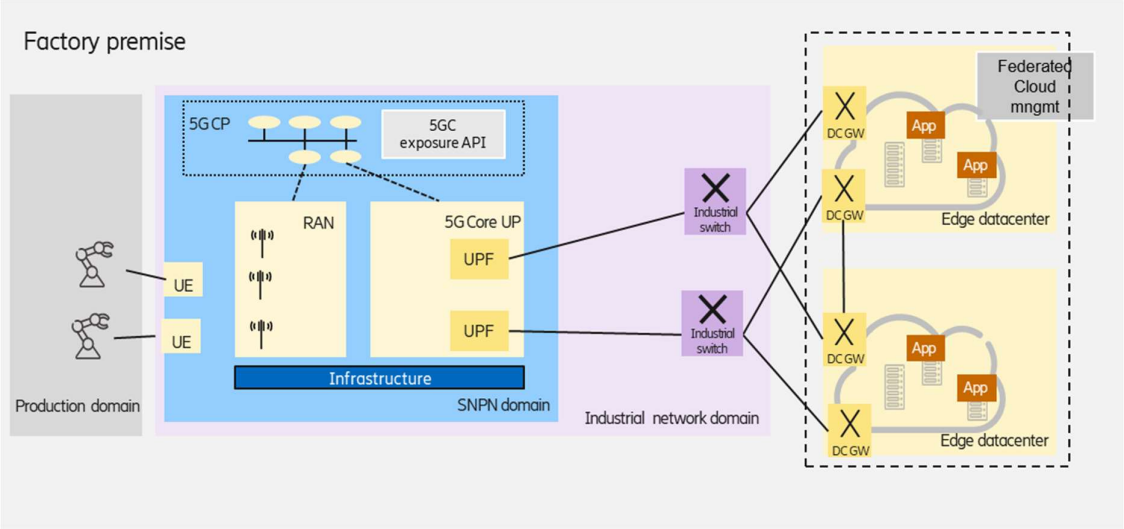


Figure 9 Federated edge datacenters in factory premise

The Federated Kubernetes allows the management of multiple edge datacenters in an integrated way [5GS20-D52]; from the enterprise customer perspective, the multiple clusters are seen as a single one. Kubernetes cluster federation is trying to solve the orchestration of clusters in the same way that Kubernetes orchestrates containers, i.e., it leverages commonly used components in Kubernetes. Connecting multiple Kubernetes clusters only requires IP reachability between the gateway nodes of the clusters. If the clusters are connected over a public network, encrypted VPN tunnel is built over the IP transport network. In case of two clusters often two independent VPN tunnels are set up for redundancy. Inside the factory premises the VPN tunnels can be omitted, as the inter cluster networking depends on the underlying network infrastructure of the edge domain. The multi-cluster features use the default (primary) networking capabilities of Kubernetes that operate at IP networking level. However, if the industrial network domain, e.g., the TSN backbone network of the factory should be interconnected directly via Ethernet with the Kubernetes cluster, then a secondary network must be set up. VPN tunnels are not applicable via the industrial network.

Integrated edge and central cloud premises

A complex industry application can consist of several software components that may have different latency, reliability, data privacy, and other requirements and the components are running in the on-premises edge and in the central cloud premises in a distributed way. The components with strict latency requirement are deployed at the on-premise edge, while components with relaxed latency

and data privacy requirements could be deployed at central (private or public) cloud premise(s) as shown in Figure 10. In this case, the cloud management should enable the handling of edge and central clouds in an integrated way, which could be treated, e.g., by KubeEdge [5GS20-D52].

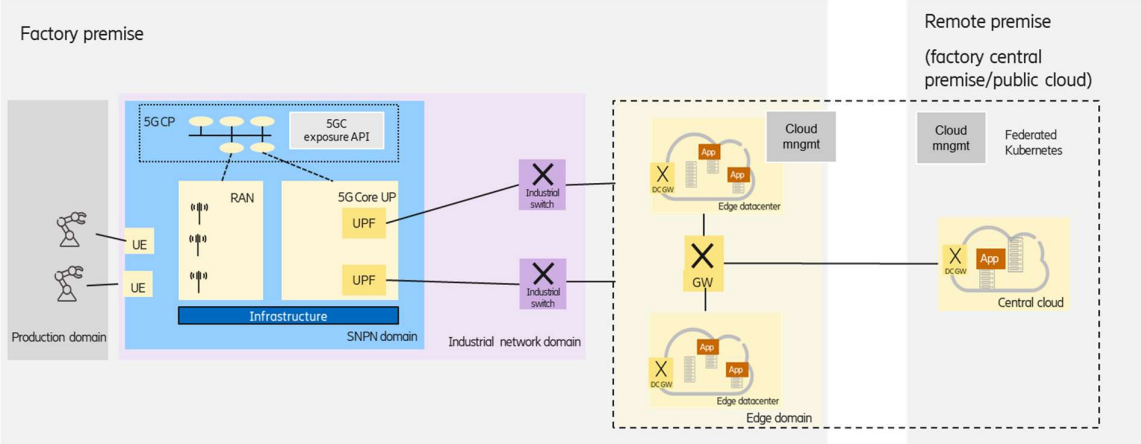


Figure 10 Integrated edge and central cloud premises

KubeEdge is an open-source CNCF (Cloud Native Computing Foundation[15]) project that extends Kubernetes to support edge computing sites and edge device management. It is based on a centralized control-plane approach[16] considering an edge infrastructure as part of the central cloud, as opposed to Kubernetes Federation, where independent clusters are united. The new architectural elements of KubeEdge provide edge computing support. The CloudHub and EdgeHub components provide message-based communication between the master and the edge-nodes over a single TCP connection. The device-controller supports the control of edge devices as well as the reporting of their status. The edge-controller is an extension of the Kubernetes controller providing event channels and orchestrating state synchronization. KubeEdge also integrates a standardized interface to discover and query edge devices from the containers.

2.3.4 On-premises edge deployed on shared NPN infrastructure

Deployment of Non-Public Networks enables that the NPN infrastructure can also host edge computing workload as shown in Figure 132, in this way the NPN can act as distributed cloud infrastructure resource, resulting the infrastructure sharing of the given NPN deployment model (e.g., SNPN) and the edge. For simplicity, the wired Industrial network segment is not shown in the figure, however the edge computing domain – as a common industrial edge deployment – can handle any industrial device which is connected to the industrial network (and not only those that are served by the 5G NPN). Albeit, this scenario has many similarities to the scenarios described in section 2.3.3, but infrastructure sharing enables some specific features, which are useful to emphasize.

[15] https://www.cncf.io/
[16] Centralize control-plane means that the cloud manager entity is deployed in the central cloud.

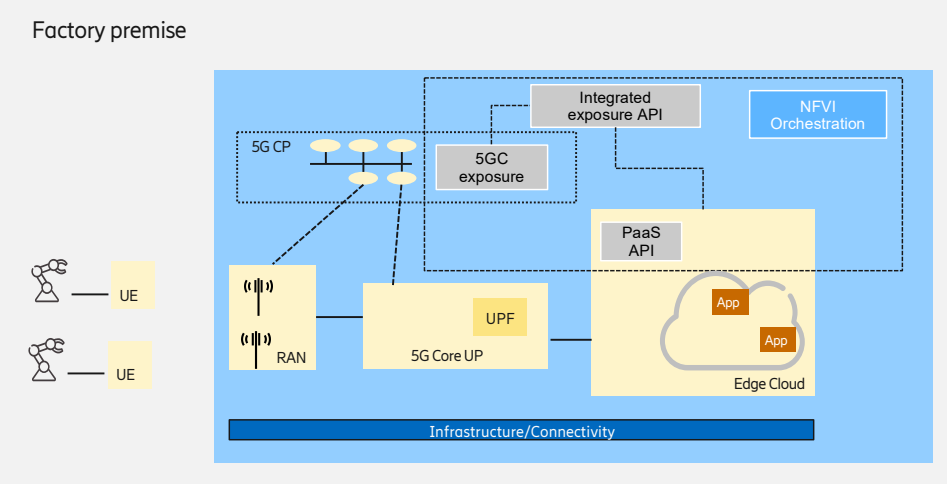


Figure 11 On-premises edge deployed on shared NPN infrastructure

This scenario enables the handling of the NPN and edge computing domains in a more integrated way, providing the following advantages for the enterprise customer (industrial party in this case):

- Common exposure API for the NPN network and edge domains, which enables a tighter interworking between domains to fulfill end-to-end requirements,
- A single NFVI (Network Functions Virtualization Infrastructure) management system could be used to orchestrate the NPN and edge resources (as a distributed cloud infrastructure), as well as Life-cycle management (LCM) is also provided,
- A 3rd party integrator can manage the edge computing (and the NPN as well) deployment on the shared, distributed infrastructure,
- Runtime execution environment is deployed and configured according to the customer needs.
- A managed, customized, full-fledged Kubernetes cluster can be offered for the customer, where the master node(s) are created, and all the required control plane mechanisms are installed. The control plane redundancy/scaling is also managed. The factory owner can then focus on the deployment and lifecycle management of its applications,
- Platform as a Service components could be the part of the solution provided by the 3rd party integrator, e.g., TSN-FRER support, time-synchronization,
- Different cloud service models can be supported according to the customer needs, such as Infrastructure as a Service (IaaS), where the customer can create VMs, containers, install the OS, etc.; Platform as a Service (PaaS), where the customer can deploy and manage its applications; as well as Software as a Service (SaaS), where the customer can directly use the installed software applications,
- Other NPN-edge domain integration solution, such as the extension of the security zones towards the edge computing domain can also be managed,
- Considering security and data privacy concerns 3rd party managed services (as Software as a Service) can also be offered to the customer.

2.3.5 On-premises edge integrated in PNI-NPN with shared RAN and core control plane

The deployment of PNI-NPN scenarios enables the MNO to provide new offerings for enterprises, such as industrial parties. In the Shared RAN and core control plane option the user plane traffic remains on-premises, while the control plane functions are hosted by the MNO public network (shared RAN and control plane).

The edge computing-related services and features that can be offered for the enterprise customer are quite similar to the listed ones in section 2.3.3, but in this case instead of a 3rd party integrator, the MNO can manage the NPN and the edge computing domains in a tightly integrated way.

Since the MNO is more involved in this scenario, this enables to move towards a hybrid scenario for higher availability. The UEs could be allowed to connect to the MNO's public network, as well as backup industry application instances can be deployed on the edge datacenters hosted by the MNO's sites near the factory premise. In the case of any on-premises user-plane NPN failure, the UEs can roam to the MNO public network and connect to the edge application on the MNO's sites.

2.3.6 Edge integrated with PNI-NPN hosted by the public network

Telco offered edge computing provides well-defined benefits for the enterprise customers leveraging the proximity to the end devices, thanks to the large footprint of an MNO. The geographical density of points of presence (e.g., Radio Access Sites) of an MNO enables the deployment of edge premises even in 10 km range from the end devices. Considering the 3GPP URLLC features combining with edge computing services provided by the MNO enables that the support of industrial use cases, which has low latency requirements, becomes realistic.

An MNO can provide IaaS for a 3rd party edge service provider that can offer the edge computing services to the market. The MNO provides the edge connectivity, compute and storage infrastructure, while the edge service provider can use this infrastructure to provide platform services to the customers (enterprise, factory). The edge service provider may offer a full commercial PaaS for the customer or could act as an IaaS provider by enabling other (cloud) service providers to integrate the MNO edge infrastructure into their cloud services, which can be offered to the customers as PaaS and/or SaaS.

Alternatively, an MNO may act as an edge service provider; in this case the MNO can deploy its own edge platform on its infrastructure (connectivity, compute, storage) and offer managed edge services (PaaS) directly to the customers.

The integrated Public network-hosted PNI-NPN and telco edge solution offered by the MNO enables that the MNO can offer a fully-fledged end-to-end solution for the manufacturing use cases, covering both the connectivity and the compute domains.

Figure 15[17] shows the case, when the edge computing service is offered by the MNO as PaaS (as mentioned above, 3rd party edge service provider can also provide PaaS for the consumer using the MNO edge infrastructure).

[17] For simplicity, industrial network domain deployed in the factory is not shown on the figure, but edge computing domain can also serve end devices that are connected to the wired industrial network segment.

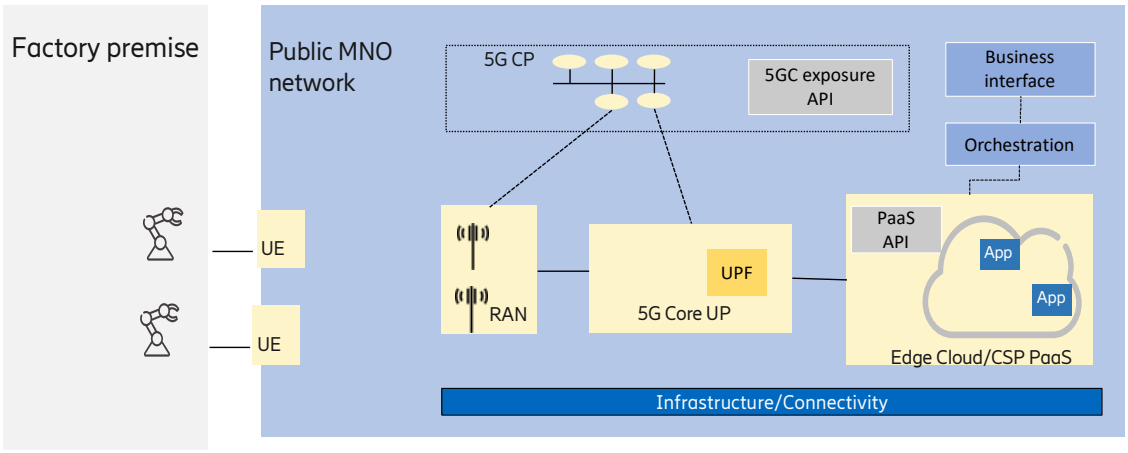


Figure 12 Edge computing service and PNI-NPN offered by MNO

The MNO can orchestrate the resources required for the PNI-NPN (which could be realized as a network slice), as well as a (customized) Kubernetes can be offered to manage the placement of the enterprise customer application workloads. The management of application instances (placement, life-cycle management) can be handled by the enterprise customer, but it also can be managed by the MNO according to a business relationship.

Depending on the footprint of the MNO, this scenario can support the case when the enterprise owns multiple factory premises. Since the offloading of application workload to the edge can be considered in many different manufacturing use case, numerous, different type application server instances should be deployed on different edge sites. The crucial point in this case is the selection of an appropriate edge premise that can serve a given industrial device according to the E2E communication service requirements. If the UE is edge-aware according to 3GPP SA6 TS23.558 [3GPP21-23558], then the interaction between the Edge Enabled Client and Server supports the fine-grained selection of an application server, considering detailed client and server profile and capability information.

Figure 13[18] shows another scenario, where the edge is deployed at the factory premise and the MNO provided PNI-NPN is used to connect the industrial devices equipped with UE to the on-premises edge.

[18] For simplicity, industrial network domain deployed in the factory is not shown on the figure, but edge computing domain can also serve end devices that are connected to the wired industrial network segment.

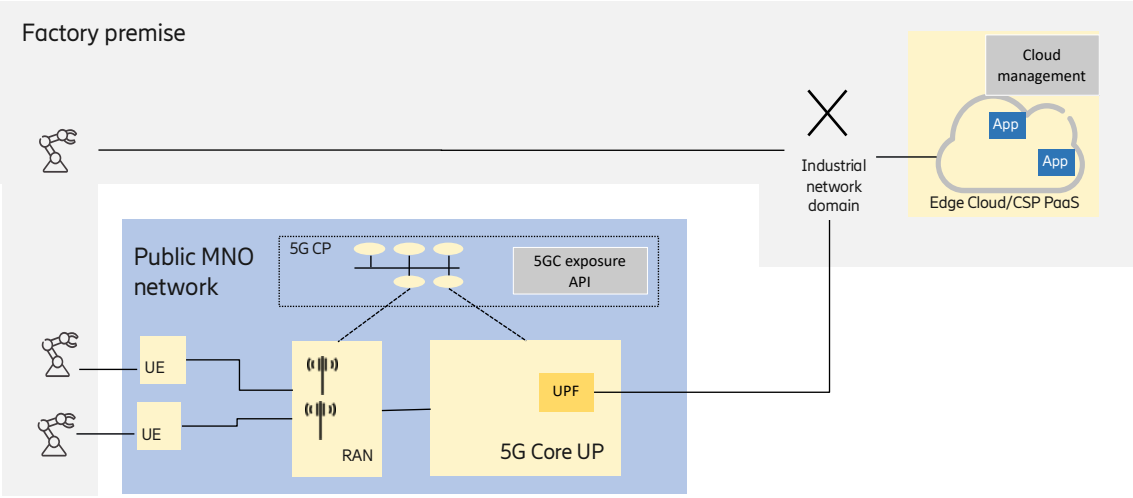


Figure 13 On-premise factory edge with PNI-NPN

The key point in the option is the proper selection of UPF that is close enough to the edge premise in order to meet the latency requirements. According to 5G system architecture specified by 3GPP working group SA2, the UPF selection can be performed by considering UE subscription and/or UE location, which is suitable for Industrial scenarios. Depending on the relationship/agreement between the MNO and the enterprise customer (Factory owner), information from the 3GPP application function (AF) entity could also be used to influence the UPF selection and traffic steering. The on-premises factory network can be considered as a local Data Network and can be reached from the UPF via N6.

2.3.7 Hybrid options

The specification of NPN options in 3GPP as well as in 5G-ACIA enables that different NPN options could be used to serve a factory deployment in a hybrid way. For example, a standalone, on-premise NPN is used for the primary communication, while an MNO hosted PNI-NPN could be applied as a backup for some critical services. Accordingly, the edge computing deployments, services could also be optimized to the hybrid NPN deployments, such as an on-premises edge can be connected to the standalone NPN and host the primary application instances, while a PaaS offered by an MNO could be applied for deploying backup application instances on an edge site close to the factory premise.

Summary on the edge computing integration with NPN deployment models

In this section an extensive analysis of the different edge computing models integrated with NPN deployments is performed. The main learnings and findings are summarized below.

The private, on-premises edge deployment integrated with SNPN deployment options enables lots of flexibility, however the edge owner (typically the industrial party in this scenario) or a 3rd party integrator is responsible for the management and maintenance of the full cloud stack (including infrastructure, virtualization, etc. domains), which may have high cost implications. In this solution wide range of edge deployment options can be supported by considering the characteristics and requirements of the different smart manufacturing applications, such as:

- Standalone edge datacenter
- Federation of edge datacenters
- Integrated edge and central cloud premises

The on-premises edge deployed on shared NPN infrastructure scenario enables the handling of the NPN and edge computing domains in a more integrated way. In this option a tighter interworking can be ensured between the NPN and the edge computing domains enabling the fullfillment of strict end-to-end service requirements.

In the on-premises edge integrated in PNI-NPN with shared RAN and core control plane scenario the user plane traffic remains on-premises, while the control plane functions are hosted by the MNO public network and since the MNO is more involved in this scenario, it enables the MNO to provide new offerings for the enterprises (e.g., industrial parties).

- The edge integration with PNI-NPN scenario enables the MNO to provide IaaS for a 3rd party edge service provider that can offer the edge services to the market. Alternatively, the MNO itself can act as an edge service provider. In the latter case, the MNO offers a PaaS directly to the customers by leveraging its own edge infrastructure.
- Kubernetes container orchestration platform supports flexible options for standalone clusters, or collaborative clusters between multiple edge computing sites and central cloud sites with centralized or distributed control.

2.4 5G-TSN integration with edge computing for enhanced reliability

The typical usage of the edge computing in smart manufacturing is to perform the complex, computation, memory and storage intensive processes. For example, various (AI/ML enabled) analytics and monitoring tasks can be executed on the edge, utilizing the cloud capabilities (e.g., scalability, robustness) and resulting in more efficient control of industry processes.

The URLLC and TSN support of the 5G networks as well as the real-time capabilities of the cloud infrastructure and platform makes it possible that even the time-critical industry device control (e.g., robot control, AGV control) functionalities can be offloaded to the edge. It means that the controller applications are virtualized and deployed in the cloud, in a container or VM environment. While TSN provides deterministic networking, to provide a deterministic computing domain within the edge computing platform the resource management and the workload scheduling must be adjusted for assuring low latency guarantees. These aspects (among others) are investigated in section 2.4.6.

Since robustness is one crucial factor for industrial use cases, the reliability of such an offloaded application can be improved both in the network and in the edge computing domain. On the network side, IEEE 802.1CB [IEEE17-8021QCB] specifies the Frame Replication and Elimination for Reliability feature of the TSN, that provides the sending of frames on multiple communication paths between a Talker and a Listener by using independent network infrastructure resources as well as provides the elimination of the duplicates as necessary. Similarly, in the edge computing domain, the robustness can be increased by deploying multiple instances of the applications on different Pods/nodes.

However, there are still open challenges to address on how high reliability functionality (e.g., TSN FRER) can be realized in an integrated 5G-TSN network and edge scenario. The current section investigates such open issues and proposes new concepts to ensure seamless interworking of the TSN FRER functionality with edge computing deployment models. The section starts with providing a brief introduction to the TSN FRER feature. Further following challenges in the edge computing domain, various architecture aspects are investigated to ensure high reliability in edge computing deployment integrated with the TSN FRER feature.

2.4.1 TSN FRER brief introduction

TSN is a set of open standards to ensure deterministic communication service over IEEE 802 networks. Various aspect of the TSN have been investigated in 5G-SMART Deliverable D5.1 [5GS20-D51]. FRER mechanism is one of the key enablers for high reliability in a TSN network (defined in IEEE 802.1CB). TSN streams (flow of the time-sensitive data over the TSN network with assigned unique identifier) are supposed to deliver their frames from the Talker (source) to Listener (destination) even in changing dynamic condition of network including transmission errors, physical breakage, and link failures. FRER provides a mechanism where packets are duplicated and transmitted over two independent paths, as illustrated in Figure 14. At the source node, TSN frame is duplicated, and a sequence number is assigned to both frames by using a sequence generation function. Further the TSN frame is transmitted over two disjoint paths. At each destination node, sequence recovery function is utilized to discard the replicas of the TSN frame by controlling the sequence numbers assigned. Today, 5GS support interworking with TSN based networks. 5GS has wide range of the feature as described in the section 2.2.1 to ensure FRER functionality in the network.

Figure 14 TSN FRER concept from IEEE report[19]

2.4.2 Challenges in the edge computing domain

Similar to network failures such as transmission errors, node failure, numerous failure events should be considered in the cloud environment that can impact a deployed service instance (e.g., failure in the container runtime environment, failure of a Pod or node). Kubernetes has built-in repair capabilities for failure handling by monitoring the Pods and if a failure occurs a new Pod will be started to take over the function of the impacted one. However, the time scale of the repair process is in the seconds (or even more) timescale, which does not meet with strict Industrial requirements (low latency, high reliability). Furthermore, albeit the active – hot standby redundancy deployment of the pods can be constructed in Kubernetes, the timescale of the failure detection and switchover is also in the seconds timescale. Hence, in order to provide the seamless end-to-end communication for applications with strict low latency requirements, *multiple, active application instances* should be

[19] https://www.ieee802.org/1/files/public/docs2017/tsn-farkas-intro-0517-v01.pdf

deployed in the cloud domain, in order to secure that in the case of any (single) failure event at least one application instance remains active and can serve the device.

In order to support end-to-end availability, the interworking of the TSN FRER and edge reliability options should also be provided by considering the cloud and network capabilities as well as the characteristics of the industrial application running on both the server and the device sides.

From the perspective of the device capabilities, two basic cases can be identified:

Scenario 1: Multiple application instance handling

In the case of multiple application instances handling, the device can handle multiple application instances (e.g., the device is able to process multiple frames that come from different application instances in a communication cycle). This option has less challenges for TSN–cloud interworking since arbitrary deployment options of the application instances in the cloud domain can be handled by the device. However, typically the application software on the device side has to be adapted to the simultaneous communication towards multiple application instances, while on the server side the continuous synchronization among the instances should be handled, so backward compatibility is limited in this case.

Scenario 2: Single application handling

In the case of single application handling, the device is capable to handle only a single application instance (e.g., the device can process only those frames that comes from a given MAC/IP address). Therefore, it is required to hide the multiple application instances deployed in the cloud domain from the device by emulating a single application instance and the TSN FRER function. In order to fulfill the above requirement, the FRER functionality as well as the Talker/Listener entities should be moved into the cloud domain (e.g., realized as virtualized functions) in order to handle the TSN and the application instance deployments in a coordinated way. This option enables to reuse the existing application software (at least on the device side), so brownfield deployment (containerization of the controller application software) is supported as well as backward compatibility can be provided (legacy software

on the device side can be used).

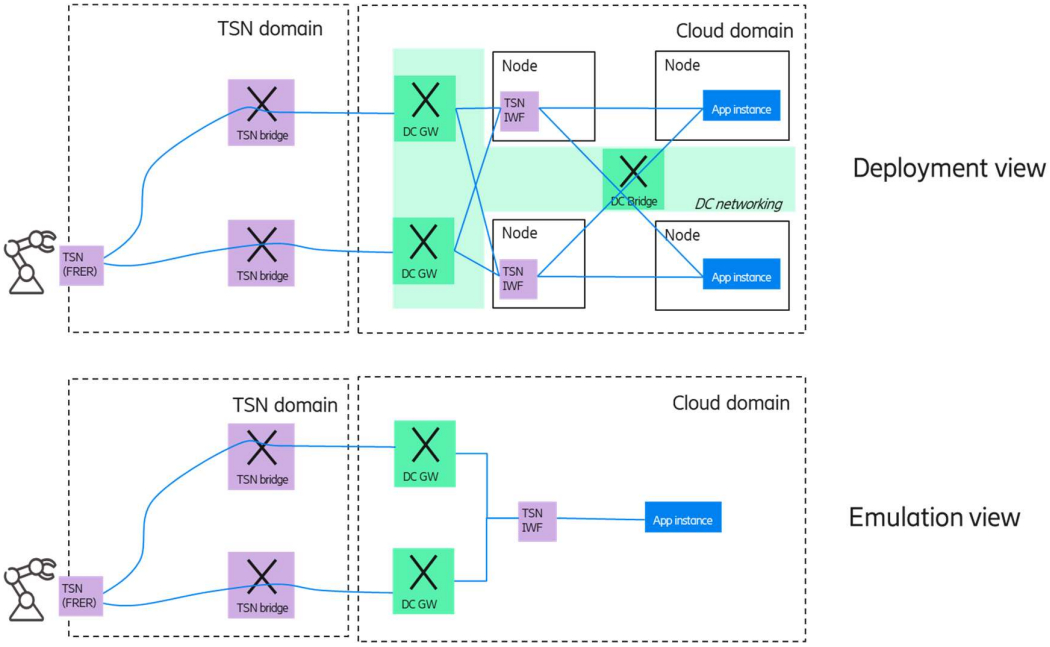


Figure 15 shows an illustrative deployment example in the cloud domain for this case as well as how the emulation should look like for the device.

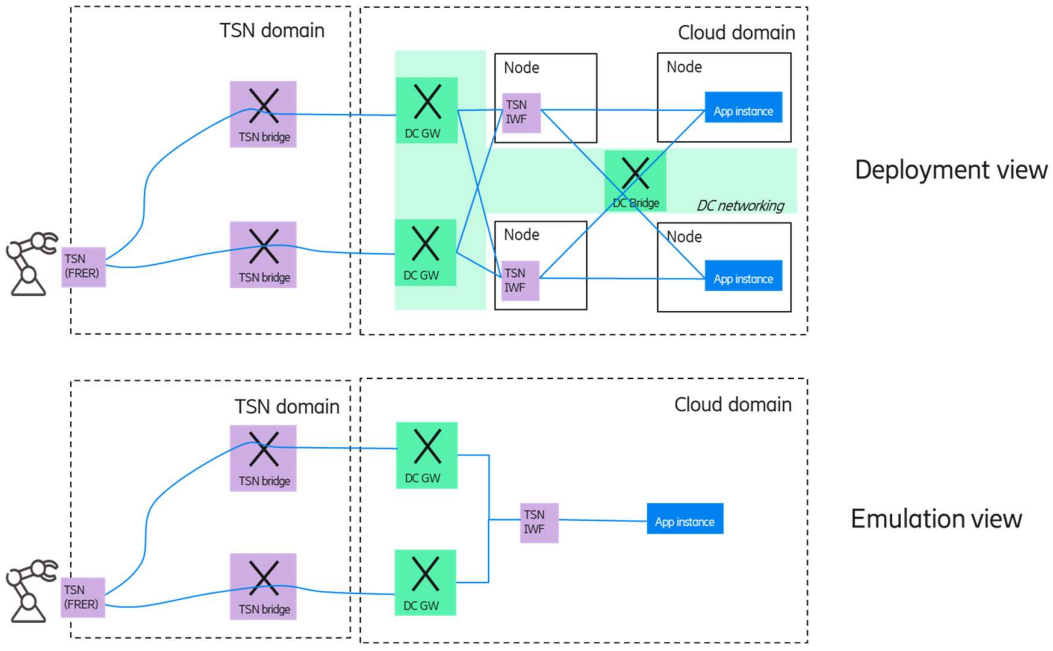


Figure 15 Single, emulated application handling scenario for TSN-cloud interworking

Proposed interworking function (IWF)

The upper part of the figure shows a possible cloud-based deployment: The cloud domain is connected to other domains via multiple datacenter gateway[20] (DC GW) nodes in order to ensure the multiple data paths between the industrial end device and the application instance(s), running in the cloud. It is also assumed that the underlying network between the datacenter nodes can offer redundant paths. To increase deployment flexibility the required TSN functions can be virtualized and comprised by a proposed new entity called as TSN Interworking functionality (TSN IWF). It includes the Talker/Listener functions as well as the FRER functionality. Optionally, other TSN features, such as IEEE 802.1Qbv (scheduled traffic), etc. can also be supported. Furthermore, TSN IWF compromises features, which are needed for emulating a single application instance towards the device – the details will be discussed in Section 2.4.4 and 2.4.5. In the deployment multiple application instances are running on different server nodes and in order to avoid single point of failure multiple TSN IWF instances are deployed. The deployment guarantees that in the case of any (single) failure in the cloud or TSN network domain, there will be an active path between the industry device and one application instance. However, when the device is not capable to handle multiple application instances, a single application and a single TSN-IWF entity should be emulated towards the device as shown in the lower part of the figure. For the device, the communication from the multiple application instances in the edge appear as two TSN FRER member streams originated from a single application instance.

2.4.3 Architectural aspects of the multiple application instances handling case (scenario 1)

Since in this case the device is able to communicate with multiple application instances, separate TSN streams (using different Stream_IDs[21]) can be established between a device – application instance pair. Per domain redundancy can be provided:

- In the cloud domain the different application instances are deployed by using different cloud resources (e.g., nodes/Pods) ensured by the orchestration system.
- The application instances are connected to different TSN IWF entities by using separate paths in the cloud domain – this can also be managed by the orchestration system if the topology of the physical network is known[22].
- In the TSN transport domain the Centralized Network Configurator (CNC) can configure different paths for the replicated frames by the FRER function. For the current integration of the edge domain the fully centralized network configuration model defined in IEEE 802.1Qcc [IEEE18-8021QCC] is assumed.

Figure shows two options for the multiple application scenario, in one case the TSN FRER is provided by the DC GWs, while in the other case, the virtualized TSN IWFs are applied. In the latter case, the TSN IWF could be realized as a Platform as a Service component.

[20] Datacenter gateway (DW GW) is the border node between the datacenter network and other network domains
[21] Unique identifier for each TSN stream (logical flow of TSN frames from Talker to listener)
[22] The redundancy level that can be achieved depends on the cloud infrastructure capabilities (e.g., number of physical servers, number of independent paths)

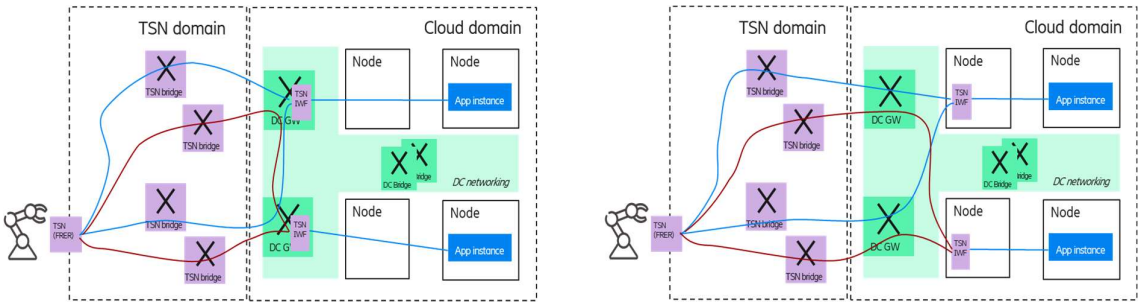


Figure 16 TSN FRER in the DC GW / TSN FRER in the virtual domain

It is important to note that due to per-stream redundancy, the required resources are scaling with the number of application instances (as the frames of all application instances are carried to the device).

The multiple application scenario can also provide redundancy without the TSN FRER since frame replication is provided by having multiple application instances. In this case – as shown in Figure 17, a single TSN Stream is established between each device and application instance pair. The usage of independent (compute and network) resources in the cloud domain can be provided by proper orchestration, while the CNC can configure independent paths for the TSN Streams over the TSN domain. In uplink direction, the device sends responses to each application instance.

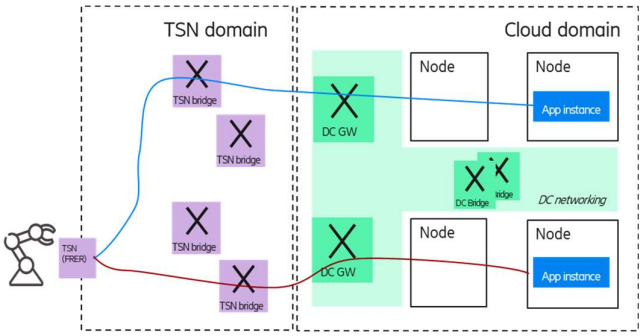


Figure 17 Per-application instance TSN stream using different transport and cloud resources

In any of the above scenarios, the 5G network can be integrated in the communication path in a seamless way as a 5G-TSN bridge.

2.4.4 Architectural aspects of the single (emulated) application handling case (Scenario 2)

This scenario is more challenging, since in this case the proper deployment of the required functions by the orchestration system is not enough, but the solution should enable the required emulation of a single application instance and TSN function towards the device. Due to the emulation, the TSN functions (FRER, Talker/Listener) should be virtualized and the TSN FRER and cloud redundancy should be handled in a coordinated way. The proposed architecture for the TSN FRER integration into cloud environment can be seen

in figure 18

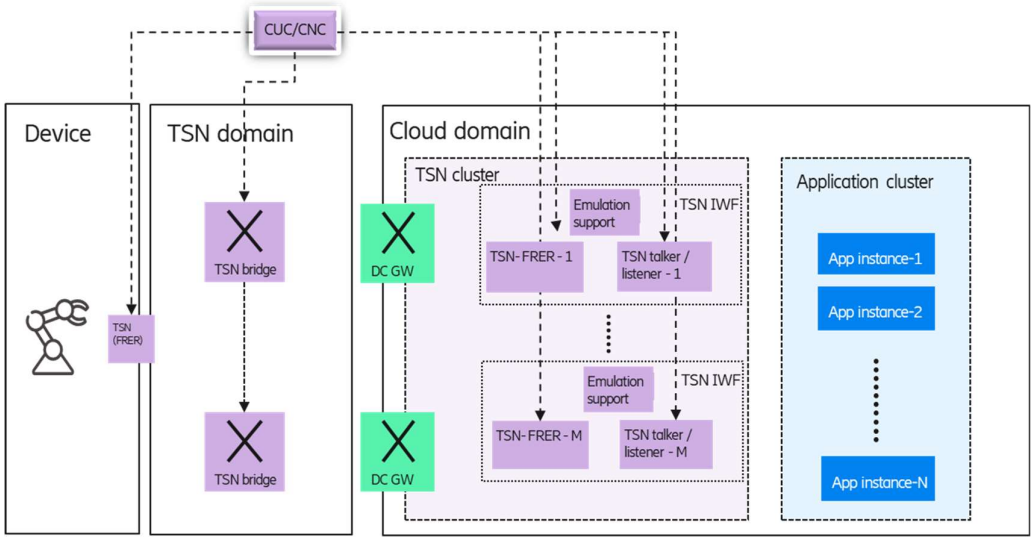


Figure 18. This section summarizes the architecture principles, the details of emulation of a single application instance are discussed in Section 2.4.5.

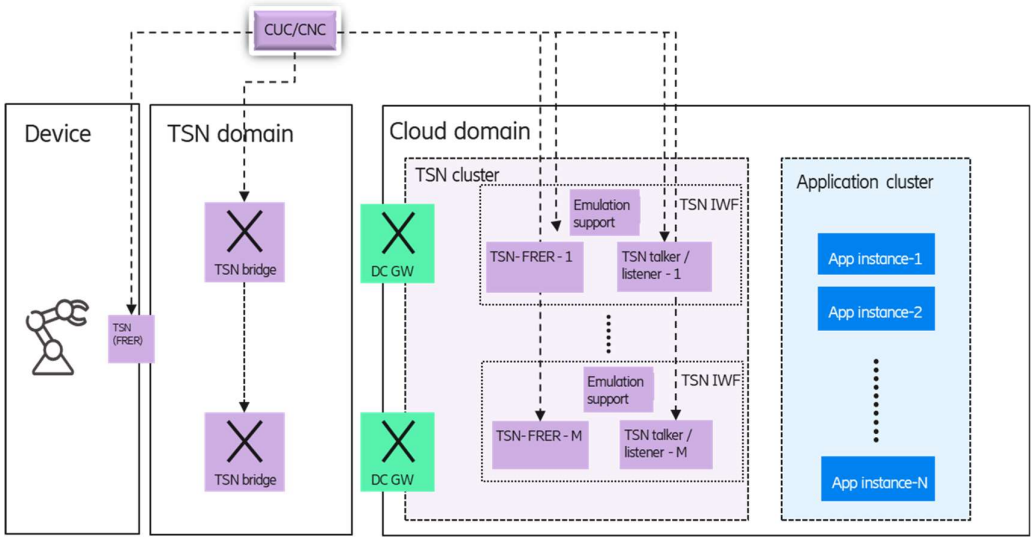


Figure 18 Architecture view of the single emulated application case for TSN-cloud interworking

The main architectural principle is to separate the management of the application instances and the TSN functions, so a separate application and TSN cluster is defined within the cloud domain. The TSN cluster (FRER, Talker/Listener functions) is configured by the TSN controller entities (Centralized User Configuration (CUC), CNC), while the life-cycle management of the application instances can be handled by using of the legacy Kubernetes orchestration features. The main reason behind this separation is to minimize the unwanted interference between the clusters, if any type of event (e.g.,

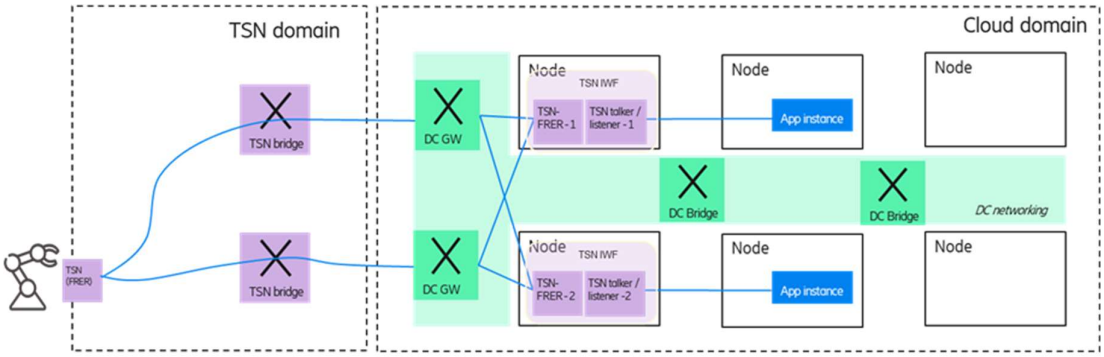
a failure) occurs in one of them. If an event (e.g., failure) occurs that impacts a TSN function, no actions are needed to perform in the application cluster. If an application instance is impacted, then the TSN configuration could remain the same (e.g., the configuration of the Talker/Listener functions, connections towards the DC GWs), so invocation of the CUC/CNC is not required[23].

Note: an option could be to deploy application instances and the related TSN functions in an integrated way, however, in this case the migration of the application instance also requires the reconfiguration of the TSN functions, so CUC/CNC has to be involved, which significantly would increase the TSN control plane actions. Furthermore, this option has no advantage considering the single application emulation aspect. Furthermore, the different TSN and application clusters enable that the number of TSN functions and application instances can be scaled independently, allowing the flexible setting of the robustness in the cloud domain.

Another architectural principle is to separate the TSN-FRER and TSN Talker/Listener virtualized components. In this way, the emulation of a single application instance could be supported by special features of the TSN Talker/Listener entities, such as IP/MAC address translation. Furthermore, the TSN Talker/Listener functions can assist the selection of the active application instance[24] in a communication cycle (for details see section 2.4.5).

Albeit the architecture enables the connection of an application instance to multiple TSN Talker/Listeners, but practically one-to-one mapping is proposed to simplify the above-mentioned coordination process. If the robustness of the system is designed to survive a single infrastructure failure in a seamless way, this simplification has no effect.

Figure 19 shows options for the single, emulated application scenario.



Option 1

[23] If an application instance fails and should be re-deployed to another node, then only the connections between the new application instance and the corresponding TSN functions are needed

[24] In a communication cycle multiple application instances can generate control messages, but only one of them will be sent to the device. In this context the "active" term means the application instance whose control message is used.

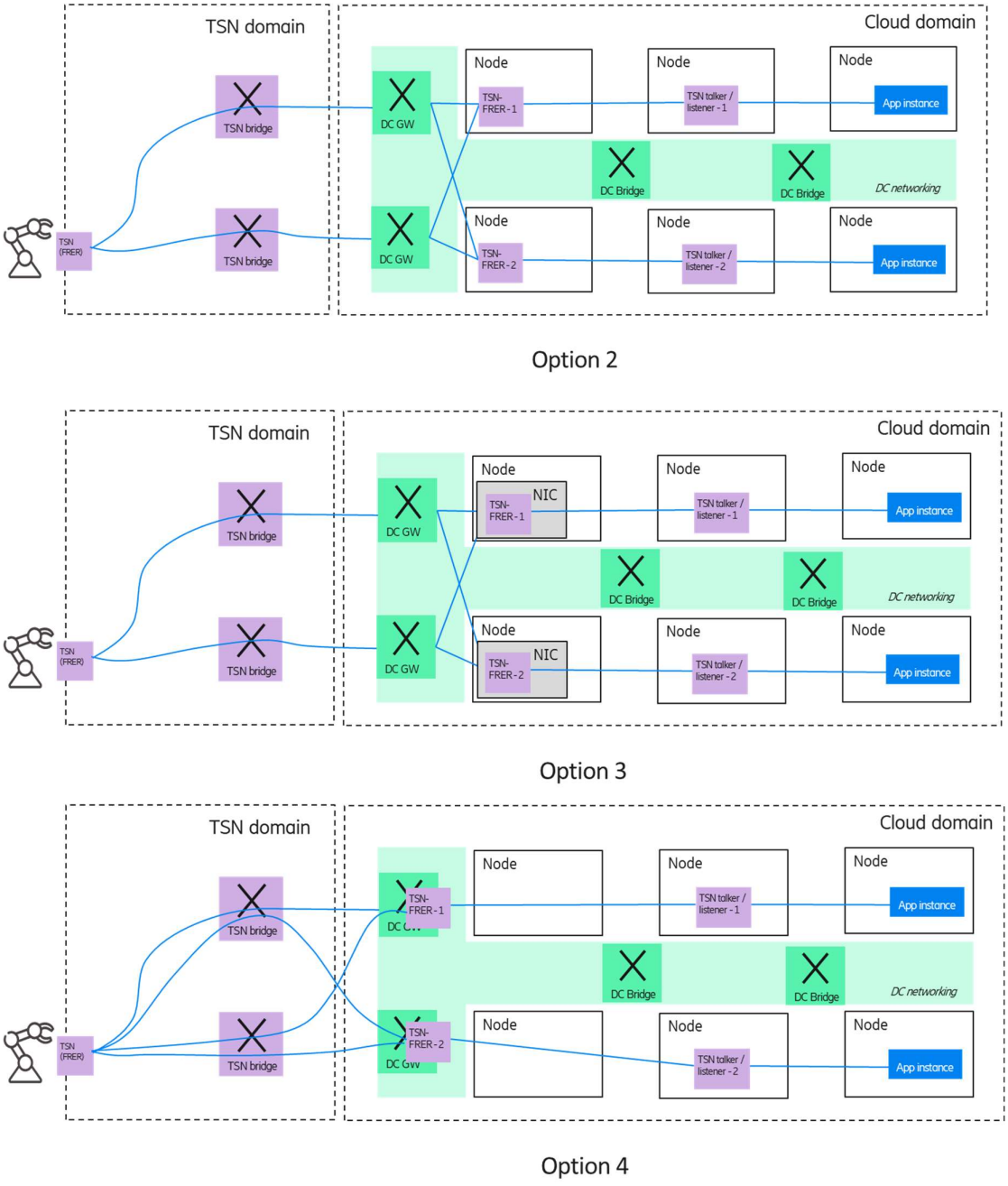


Figure 19 Deployment options for single, emulated application scenario for TSN-cloud interworking

Option 1 shows an example when the TSN Talker/Listener and FRER functions are integrated as a TSN-IWF and it is deployed in a single Pod. In the case of option 2, separated Talker/Listener and FRER functions are used, which could be deployed as different Pods on different nodes. The third option shows a case, when the TSN FRER function is deployed on the Network Interface Card (NIC) of a node as a virtual switch. Alternatively, the separation of the Talker/Listener and FRER functionality enables

that only the Talker/Listener functionality is moved to the virtualized domain, the TSN FRER functionality is provided by the DC GWs – this is shown as option 4.

2.4.5 Details of the emulation of single TSN-FRER and application instance

This section discusses the details of how the complexity of the application deployment is hidden from the device and a single application instance is emulated.

One part of the emulation is to guarantee that only one frame is sent to the device in a communication cycle; by other words, it means that selection of one application (as well as TSN Talker/Listener and FRER) instance is needed, which is used as a serving instance[25]. The selection is not required to be performed in each communication cycle, so one way is to assign one application instance as the primary one and it serves the device for a while. If any planned switching between the application instances is scheduled or any extraordinary event (e.g., a failure) occurs, then another (secondary) backup application instance takes over the device control.

If the selection of the serving application instance is managed in the application cluster, the application software should be aware of those multiple instances. The instances should have the capability to discover themselves and communicate with each other in order to perform the selection and handle the case if the serving instance cannot work anymore.

Another alternative is to handle the selection by the support of the TSN IWF entities. In this case, the application instances can be agnostic to the selection coordination. Depending on the currently selected application instance, the corresponding TSN Talker sends the frames to the device (through the FRER function), the messages coming from the other application instance(s) are blocked by the other TSN IWF entities. If the serving application instance cannot work anymore, then the other TSN IWF entities, which serve the application deployment will be informed (e.g., the cloud management can handle it or the TSN IWFs can automatically recognize each other). Then the selection coordination functionality of the TSN IWF instances is applied[26] and the message coming from the newly selected application instance is started to be sent towards the device through the corresponding TSN Talker entity.

In addition to the application coordination, the emulation of a single TSN FRER entity is also required, which requires extensions to the existing IEEE 802.1CB TSN FRER operation. The issue is that the Replication function of the FRER uses a sequence number parameter (GenSeqNum) to identify the duplicated frames. The existing IEEE 802.1CB specification does not allow the free modification of the ”GenSeqNum” parameter. However, if a change of TSN FRER instance is needed in the cloud (virtualized) domain then the ”GenSeqNum” parameters of the new and the old FRER instances will not be coordinated, which leads to unnecessary frame loss.

In order to resolve the above issue following improvements are discussed in the IEEE TSN working group:

[25] Due to fast failover, all application instances generate the control message, but only one of them will be sent to the device.
[26] The selection method is out of the scope of this document, wide range of approaches could be applied from the simple pre-configured option to the automated selection based on e.g., performance metrics

- Allow modification of the ”GenSeqNum” parameter to any valid value in the ”BEGIN” event, which is the global event that resets all FRER functions
- A new event called ”SEQUENCE_CHANGE” is proposed, which could be triggered via external entities or management intervention. In these cases, the ”GenSeqNum” is set to a specific provided value.

By using the above modifications, the seamless change between TSN FRER instances deployed in the virtualized domain can be enabled.

2.4.6 Kubernetes Capabilities for Industrial Edge Cloud

General cloud platforms that edge computing sites can be built on are not prepared for industrial applications that have special requirements concerning low latency or high reliability, because of performance uncertainties and capability gaps of these cloud platforms. These aspects have to be taken into account, as we detail them in the following, showing what kind of solutions can be applied concentrating on Kubernetes based deployments.

2.4.6.1 Infrastructure Accelerators

Infrastructure accelerators are hardware devices that provide specialized functions either to guarantee quality of service measures or offload some work from the CPU; examples are FPGAs, GPUs or (Smart)NICs. Industrial applications with real-time and latency-sensitive requirements make such accelerators more important in edge computing environments. To be able to use them in a virtualized edge cloud, mechanisms are needed to bypass system software and the virtualization platform for directly exposing the hardware to the applications.

Kubernetes can handle such hardware devices with a device plugin framework; however, it is only in beta stage. Its operation is to advertise system hardware resources towards pods (e.g., smart/NICs, SR/IOV devices, GPUs) and requires that the hardware vendor must implement the device plugin.

To run TSN functions on a general server, specialized NICs might be required to guarantee bounded latency, synchronization and jitter for time-sensitive traffic. As it was shown in the TSN-FRER architecture, the application and the TSN functions are split, therefore only some nodes of the edge cloud must have these NICs. Those nodes can be labeled, and Kubernetes provides mechanisms for the placement of the application components presented in the following subsection.

2.4.6.2 Placement of Application Components

A Kubernetes *deployment* object allows to specify, among others, which container images to use for the application, the number of replicas for the pods, and which rules for the placement related to the nodes and pods have to be followed. For example, the number of replicas is two for each of the TSN and application components in all samples presented in Figure 19.

The placement rules are specified by affinity and anti-affinity rules, that can relate to nodes and pods, too and this is a mature feature in Kubernetes. To assure that TSN-FRER pods are deployed into nodes that are labelled to have the special NIC capability, a node affinity rule is applied in the deployment description (see Figure 19, Opt. 3).). In addition, a pod anti-affinity rule is also specified such that a TSN-FRER pod cannot be placed on a node that already runs another TSN-FRER pod, to assure node level resiliency. With appropriate node and pod affinity rules the desired placement can be specified

for all cases, and the Kubernetes pod scheduler will place the pods accordingly, by selecting among the available nodes that satisfy them.

2.4.6.3 Resiliency and Healing Methods in Kubernetes based Edge cloud solution

General cloud platforms are designed for applications in which longer interruptions are tolerable, such as several minutes of outage per month, but this is certainly not the case for all kind of industrial applications. Applications with time critical control loop require low latency operations from the edge cloud too, however, some application components, such as analytics or data acquisition, do not have such strict restrictions, and therefore can utilize built-in Kubernetes mechanisms for resilience.

In Kubernetes, as a general cloud platform, a Kubernetes *service* object provides load balancing between the multiple pods that serve as the backing endpoints of that service, therefore the incoming requests are distributed randomly among them. If a pod fails, and it is detected, it will be removed from the list of endpoints, but the other pods still can serve requests. This operation provides a certain level of resilience.

For application components without strict low latency requirements the hot standby resilience can be a satisfactory solution that utilizes built-in Kubernetes mechanisms. The hot standby operation among two pods in a service is not supported natively; however, it can be constructed by using leader-elector sidecar containers combined with readiness probes assuring that only one pod out of the two is in the ready state. Therefore, the load balancer can forward messages only to this single active pod. If the failure of the single active pod is detected, then the other pod will change its state to ready and take over the duty. As soon as the failure of the first active pod is detected by the Kubernetes platform, a new pod will be started after some time to replace the old one and will now serve as hot standby pod. As the minimum interval for the readiness probes is 1 second, the failure detection and switchover time is also in this order of magnitude.

Therefore, for application components with strict low latency requirements below the seconds order of magnitude the active-active application is desirable for seamless resiliency. To guarantee industrial grade resilience with a very short switchover time the duplication of the application components is needed, and they have to be operated in the active-active model. To map this operation into Kubernetes objects, separate *deployments* and *services* have to be constructed for the individual application and TSN functions, running single pods for each active instance. Within the deployments, the standard Kubernetes respawn mechanism will restore the pods in case of failure, but at the application level, because of the duplication, the device is always controlled by at least one application instance. If the device can handle multiple application instances, it can connect to these separate services, if a single application instance must be emulated then the duplication is hidden and the applications or the TSN Talker/Listener entities make the selection among the services.

Kubernetes built-in mechanism are used for fault recovery for pods that are part of a deployment. The status of the pods is monitored by the Kubernetes system, and if a pod fails, another identical pod will be launched. This is a reactive respawn, and can have quite a long service interruption, until the new pod is available to serve requests. However, this feature can be used for automatically restoring the failed application components in the active-active resilience model too, to ensure multiple active components at the end of the recovery process. This method is ideal for stateless applications. However, for stateful applications where the state has to be restored in the new pod instance, the

application has to handle this and must be aware of restarts and state restorations, and this also can add time to the service interruption. To store the state, Kubernetes persistent storage or some third-party database application is also required.

2.4.6.4 Kubernetes networking aspects

Networking requirements are about facilitating connectivity between attached devices and edge applications. The requirements on connectivity typically vary between different types of edge applications. Kubernetes natively provides Layer 3 IP traffic handling within the cluster and from external hosts to services. The Kubernetes network model assigns IP addresses to pods and services. By default, a Kubernetes pod has only one network interface, and all traffic goes through this interface, such as communication between the Kubernetes API and the pod, Kubernetes probes for liveness and readiness and the user traffic. However, this default single pod network interface interconnected with the Kubernetes cluster networking is not appropriate for directly connecting to an external TSN network segment to forward the Layer 2 TSN traffic directly to a pod. Kubernetes has the option to attach multiple network interfaces to pods, that can be attached to a different network. This feature is provided by the multus Container Networking Interface (CNI) plugin. Multus is a meta-plugin in the sense that it can call multiple other CNI plugins for the different interfaces. With multus, a secondary network interface for the TSN traffic can be defined for a pod in the pod specification. For this secondary interface another CNI plugin, the macvlan plugin can be used. The macvlan plugin functions like a switch that is already connected to the host interface of the node the pod is running on. These virtual interfaces share the physical network device of the host but have distinct MAC addresses. The nodes in the Kubernetes cluster that are designated for receiving TSN traffic are to have a secondary physical NIC for this purpose, for sharing towards the secondary interfaces of the pods and connect to the TSN network.

Network Service Mesh is another, more abstract level initiative to extend the networking capabilities of Kubernetes. It allows heterogeneous network configurations and on-demand, dynamic, negotiated connections with minimal need for changes to Kubernetes. It extends the Kubernetes API with functions to facilitate connectivity between containers running services or with external endpoints, and the payload type can be Ethernet or MPLS in addition to IP. It also provides resiliency as it can auto heal connections between pods and network services if various system elements restart or if the network service fails without disturbing client pod. Unfortunately, the project seems to be not active since around one year.

2.4.7 Kubernetes Resource Management aspects for low-latency workloads

While the deterministic behaviour of the network can be provided by 5G/TSN features, cloud services generally do not provide guarantees and can exhibit non-deterministic performance due to shared compute and network resources.

For production workloads the Kubernetes resource management must be understood by its operator. In a pod specification it can be optionally specified how much of each resource a container requires. The most common resources to specify are CPU and memory (RAM). When the resource request is specified for containers in a pod, the scheduler uses this information to decide on which node to place the pod and also reserves at least the requested amount of that system resource specifically for that container to use. When a resource limit is specified for a container, those limits are enforced so that

the running container is not allowed to use more of that resource than the limit set. If the node where a pod is running has enough of a resource available, it is possible (and allowed) for a container to use more resource than its *request* for that resource specifies. However, a container is not allowed to use more than its resource *limit*. In case of CPU limit the pod will be throttled if it exceeds its limit and can be evicted if exceeds the memory limit[27]. To properly set *requests* and *limits* the resource usage of the application must be known by measurements for example.

The low latency operation of a container can be affected by the CPU resource settings. *Limits* and *requests* for CPU resources are measured in CPU units. One CPU, in Kubernetes, is equivalent to **1 vCPU/Core** for cloud providers and **1 hyperthread** on bare-metal processors and fractional units are allowed, such as 500m meaning 500 millicore, i.e., half CPU.

When the *request* is set to a value less than the *limit* then the scheduling decision is made by taking into account the request. The requested resource amount is guaranteed for the container, but the container is allowed to use resources up to the specified *limit*, if the node has enough free resources. Pods with such containers are in the burstable QoS class. If both for CPU and memory resources, the *requests* and *limits* are set to the same values for all containers in a pod then it is in the guaranteed QoS class.

The CPU *request* value is used by the Kubernetes pod scheduler, however the CPU limit value is enforced by using the Completely Fair Scheduler (CFS) that is the default process scheduler in Linux for normal tasks that have no real-time execution constraints. CFS CPU bandwidth control is a kernel feature on the host that runs the containers and allows the specification of the maximum CPU bandwidth available to a group or hierarchy of processes. The *limit* value for a container is enforced by the CFS, for all processes running inside the container, i.e., group of processes. The bandwidth allowed for a group is specified by using a quota and period. Within each given “period” (microseconds), a group is allowed to consume only up to “quota” microseconds of CPU time. When the CPU bandwidth consumption of a group exceeds this limit (for that period), the tasks belonging to its hierarchy will be throttled and are not allowed to run again until the next period starts.

When the container CPU limit is set, the resulting value is converted to its millicore value and multiplied by 100. The resulting value is the total amount of CPU time that a container can use every 100 ms. A container cannot use more than its share of CPU time during this interval. The default quota period is 100 ms, and the minimum resolution of CPU quota is 1 ms, this can be set on the host level. If the application running in the container realizes a periodic control loop, then the quota period is to be adjusted in accordance with the periodicity of the control loop for the control process to be scheduled for each control time period. Still, CFS is for normal tasks that have no real-time execution constraints. In the Linux kernel there are other schedulers available for real-time scheduling: real-time first-in-first-out, real-time round-robin and deadline scheduler. Currently the usage of these schedulers is not implemented in Kubernetes.

Unfortunately, there is a negative side effect of CPU limits, as the CPU limit is enforced by restricting the total amount of CPU time that a container can use every 100 ms. For example, if the limit is set for 400m then the container can run 40 ms in each 100 ms time window, however, when a request is not

[27] https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/

processed within 40 ms, then 60 ms waiting (throttling the process) will prolong the response time and this can happen several times, until the request processing is finished. Unfortunately, because of a Linux kernel bug that is fixed only in kernel version 4.19, a container can be throttled even without the CPU usage getting close to the limits. Because of this, it is recommended to define no CPU limits or to disable the enforcing of CPU limits in Kubernetes platform level, but this can be done only in a self-managed cluster. In addition, without limits no prevention mechanisms are provided by the Kubernetes platform and alternative ways are required to prevent high CPU usage for pods, such as monitoring the CPU usage and adjusting the requests accordingly.

However, there are Kubernetes tools that support better performance isolation for selected pods to serve workloads sensitive for, e.g., CPU throttling or context switches. The *CPU manager*[28] is a beta feature in Kubernetes that can allocate exclusive CPUs to certain pod containers. The pod must be in the guaranteed QoS class and whole numbers of CPU cores must be specified in the request and limit, e.g., 1000m or 3000m, to allocate exclusive cores. This way the containers do not share the CPU resources and as a result, better performance is expected.

To provide low latency performance enhancements in a Kubernetes platform further low latency features can be configured on the nodes of the cluster, this, of course, supposes a self-managed cluster. These cover hardware settings and tuning of the software on the nodes, especially the Linux kernel. For best response times, it is recommended to disable power management options in the BIOS, as various CPU sleep states can affect how quickly the system responds to external events. Another option is to update the kernel to kernel-rt, that is an optimized kernel designed to maintain low latency, consistent response time, and determinism in contrast with the normal one, that focuses on throughput-oriented operations and fair scheduling of tasks. The optimized nodes are to be labelled and node selectors for the pods will ensure the placement on them. The CPU cores of the node can also be partitioned to serve Kubernetes management processes in one partition and to serve latency sensitive workloads in another partition not to interfere with each other.

2.4.8 Security zones enabled in Edge cloud

Industrial applications and devices are typically clustered into security zones. Network security zones provide network segmentation by breaking down the network into physical or logical zones with similar security requirements. [5GS20-D52 section 5.6.1]

This section provides details on the extension of the work towards security zone in an edge cloud integrated in an 5G based Ethernet network. To separate network traffic within the Kubernetes cluster network policies can be applied. In Kubernetes, pods can communicate with each other and will accept traffic from any source, by default. Kubernetes network policies allow to control traffic flows at the IP address or port level. These policies are application specific and restrict how a pod can communicate with other network entities. The allowed communication can be specified by three identifiers: by other pods, namespaces and IP address blocks. Labels are used to select pods and specify the traffic that is directed toward those pods using rules. To use network policies, a capable Kubernetes network plugin must be used. Most Container Network Interface (CNI) plugins (e.g., Weave, Calico, Cilium, etc.)

[28] https://kubernetes.io/blog/2018/07/24/feature-highlight-cpu-manager/

support the implementation of network policies, however, if they do not and a "*NetworkPolicy*" is yet created, then it will be ignored.

A straightforward solution is to map the factory security zones to Kubernetes namespaces and set up the network policies for namespaces according to the security zones, such as the namespaces are isolated, i.e., pods can communicate with pods in the same namespace only.

Summary for the edge computing integration with 5G-TSN based industrial networks

This section summarized the architecture, design and implementation aspects of edge computing integration with 5G-TSN FRER in order to ensure an end-to-end, integrated reliability solution. The main concluding remarks are as follows:

From device capability perspective two alternatives can clearly be identified: 1) the industry device can simultaneously handle multiple application instances, and 2) the industry device can handle only a single application instance.

Regarding the multiple application instances handling scenario the main identified findings are:

- This options fits to greenfield deployment or for such cases when the device software is written in a way to handle multiple application instances. The main drawback of this option from the viewpoint of legacy deployment is that the industrial end device software has to be adapted to the simultaneous communication towards multiple application instances.
- In this case the end-to-end reliability can be provided on a per-domain basis by properly configuring
    - the placement of application and TSN FRER instances in the edge domain, ensured by the cloud orchestrator.
    - The disjoint paths for the TSN streams in the TSN domain, ensured by the CNC.
- In this scenario, reliability can be provided even without TSN FRER; in such case each stream between the device and an application instance has to be configured to use independent path in the TSN domain by the CNC.

Regarding the single application instance handling scenario, the main identified findings are:

- This option does not require new capability on the device side, so it provides backward compatibility and fits for legacy/brownfield deployments.
- On the other hand, the device capability (e.g., single application instance handling) requires tight interworking of the TSN FRER and edge computing domain in order to hide the multiple application instance from the device:
    - For the most efficient interworking, the TSN FRER functionality should be virtualized and moved into the cloud domain.
    - Coordination for selecting the active application and TSN FRER instances as well as the capability of seamless application instance change is a must.
    - Improvements are needed in the TSN FRER operation, currently discussed in the IEEE TSN working group.

Kubernetes can be used for containerized applications (or VMs with the help of KubeVirt add-on) as the edge computing platform integrated with 5G-TSN industrial networks, as it has several mature

functions that support this kind of integration, such as the placement of application instances according to the redundancy requirements, support for dedicated secondary network attachments for application instances towards the TSN network and industrial security zone mapping within the Kubernetes cluster. Also, there are some immature capabilities that seem to need some improvement, but are promising for industrial use cases, such as the separation of latency sensitive workloads or the support for direct hardware access from the application containers.

2.5 QoS management

From the network architecture perspective as indicated in previous 5G-SMART Deliverable D5.2 [5GS20-D52], QoS is one of the 3GPP technical enablers. In this report, we investigate general mechanism of enabling QoS for smart manufacturing application. 5GS defines a QoS framework to satisfy diverse application QoS requirements. The 5G QoS model is based on QoS Flows. The QoS Flow is the finest granularity of QoS differentiation in the 5GS. A QoS Flow ID (QFI) is used to identify a QoS Flow in the 5G System. User Plane traffic with the same QFI within a Packet Data Unit (PDU) Session receives the same traffic forwarding treatment (e.g., scheduling, admission threshold).

Some characteristics of the 5G QoS model are:

- The model is E2E and provides the necessary hooks at the UE, Radio Access Network (RAN) and the UPF levels
- A Service Data Flow (SDF) is the term used in 3GPP for an E2E packet flow. SDFs are mapped to QoS Flows by the UPF for downlink data and by the UE for uplink data
- User plane marking for QoS is carried in encapsulation header without any changes to the E2E packet header
- Applies to both IP and Ethernet PDUs

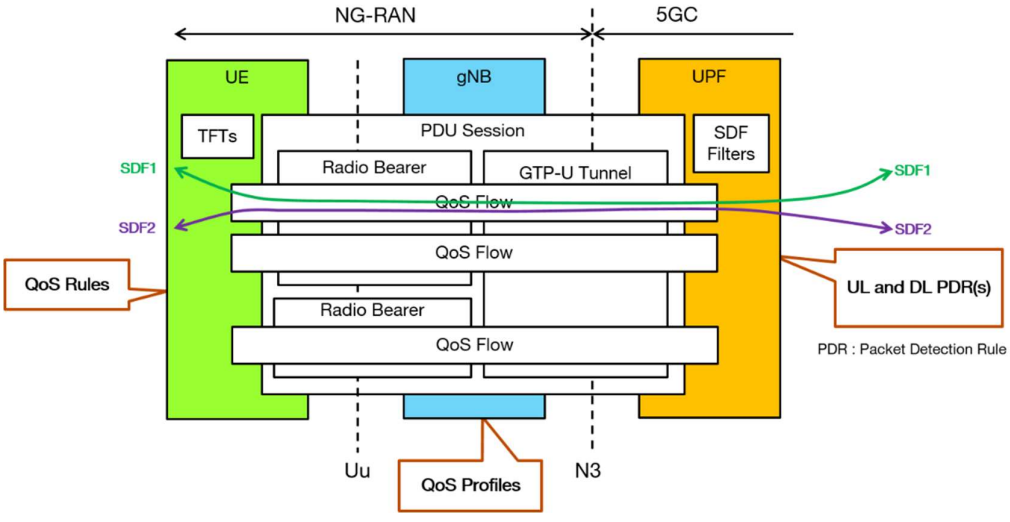


Figure 20 Generic view of QoS architecture

Several new terms such as QoS Rules and QoS Profiles have been introduced in the above figure and later in this section these terms will be explained. This figure illustrates some of the characteristics of

the model – multiple QoS flows are possible in a single PDU session, mapping of SDF to QoS flow, E2E nature of QoS support with the involvement of different 5G network functions / network elements.

The differentiation that is provided by a QoS flow is characterized by the following 5G QoS parameters and notification:

5QI - The 5G QoS Identifier is a scalar used as a reference to the 5G QoS characteristics of a QoS Flow. 5QI QoS characteristics may be standardized and/or pre-configured values or dynamically assigned 5QIs

ARP - Allocation and Retention Priority (1 to 15) and pre-emption capability/vulnerability of the QoS Flow. ARP is used to prioritize resource allocation to QoS Flows.

Reflective QoS Attribute (RQA) - Reflective QoS refers to the ability to use downlink information on QoS to setup the uplink QoS.

Notification control - For GBRs (described below) when the QoS characteristics cannot be satisfied, a notification can be sent by the NG-RAN. A notification is also sent when the RAN can once again satisfy the QoS.

Guaranteed Bit Rate (GBR) flows meet the corresponding QoS characteristics targets provided the traffic flows are within the data rate (GFBR) and data bursts are within the limits specified in Maximum Data Burst Volume. Non-GBR flows are best effort in nature.

The 5G QoS characteristics and notifications are:

- Resource Type (GBR, Delay Critical GBR or Non-GBR)
  - Dedicated resources are permanently allocated to GBR or Delay Critical QoS Flows.
  - For GBR QoS Flows with delay critical resource type, a packet which is delayed more than PDB is counted as lost, and included in the PER.
- Priority Level
  - It indicates a priority in scheduling resources among QoS Flows of the same UE or QoS Flows from different UEs.
- Packet Delay Budget (PDB)
  - It defines an upper bound for the time that a packet may be delayed between the UE and the ingress/egress point at the UPF.
  - The PDB is composed of the 5G Access Network Packet Delay Budget (5G-AN PDB) and of the CN Packet Delay Budget (CN PDB).
- Packet Error Rate (PER)
  - It defines an upper bound for the rate of PDUs (e.g., IP packets) that have been processed by the sender of a link layer protocol (e.g. Radio link control (RLC) in RAN of a 3GPP access) but that are not successfully delivered by the corresponding receiver to the upper layer (e.g. Packet data convergence layer (PDCP) in RAN of a 3GPP access).
  - For GBR QoS Flows using the Delay-critical resource type, a packet delayed more than PDB is counted as lost if the data burst is not exceeding the MDBV within the period of PDB and the QoS Flow is not exceeding the GFBR.

- Averaging Window
  - The Averaging Window is defined only for GBR QoS Flows. It represents the duration over which the Guranteed bit rates shall be calculated (e.g., in the RAN, UPF, UE).
- Maximum Data Burst Volume (MDBV) – for GBR QoS Flows with Delay-critical resource type only
  - The MDBV denotes the largest amount of data that the RAN is required to serve within a period of 5G-AN PDB.

Table 4 Example of standardized 5QI-to-QoS mapping

5QI Value	Resource Type	Default Priority Level	Packet Delay Budget	Packet Error Ratio	Default Maximum Data Burst Volume	Default Averaging Window	Example Services
4	GBR	50	300 ms	10^{-6}	N/A	2000 ms	Non-Conversational Video (Buffered Streaming)
6	Non-GBR	60	300 ms	10^{-6}	N/A	N/A	Video (Buffered Streaming) TCP-based (e.g., www, e-mail, chat, ftp, p2p file sharing, progressive video, etc.)
…	…	…	…	…	…	…	…
82	Delay-critical GBR	19	10 ms	10^{-4}	255 bytes	2000 ms	Discrete Automation (see TS 22.261
83	Delay Critical GBR	22	10 ms	10^{-4}	1354 bytes	83	Discrete Automation (see TS 22.261). V2X messages (UE – RSU Platooning, Advanced Driving: Cooperative Lane Change with low LoA. See TS 22.186 TS 23.287)

3GPP has defined standardized 5QI values that map to certain QoS characteristics in [5GS20-D52]. Table 4 shows an extract from the table in above reference in order to illustrate the above characteristics with concrete values.

Figure 21 shows the architecture for end-to-end QoS. An application service configures the Application flows that are needed based on the different traffic that the services need. This is expressed in terms of SDFs at the Policy Control Function (PCF) level. Figure 21 illustrates this is as being communicated by the Application Function (AF) for the purposes of illustration. Other possibilities are for the NPN user to communicate his/her needs to the NPN operator in the context of the Service Level Agreement (SLA). It is then up to the NPN Operator and NPN integrator to setup the necessary configuration during the deployment phase.

Schematically, from the SDF description the PCF derives the 5G QoS parameters. The PCF with the help of the AMF and SMF it distributes these rules to the UE as QoS rules, to the UPF as Packet data Rules (PDRs) and to the RAN as a QoS profile. The two end points of the 5G communication (i.e., the UE and UPF) use this information to classify the application traffic and map it to the corresponding

QoS Flow. Based on the QoS flow characteristics the access network will make the appropriate scheduling and resource allocation decisions.

Note that it is also possible to preconfigure the QoS profile on the RAN instead of receiving it from the SMF (via AMF) as shown in Figure 22.

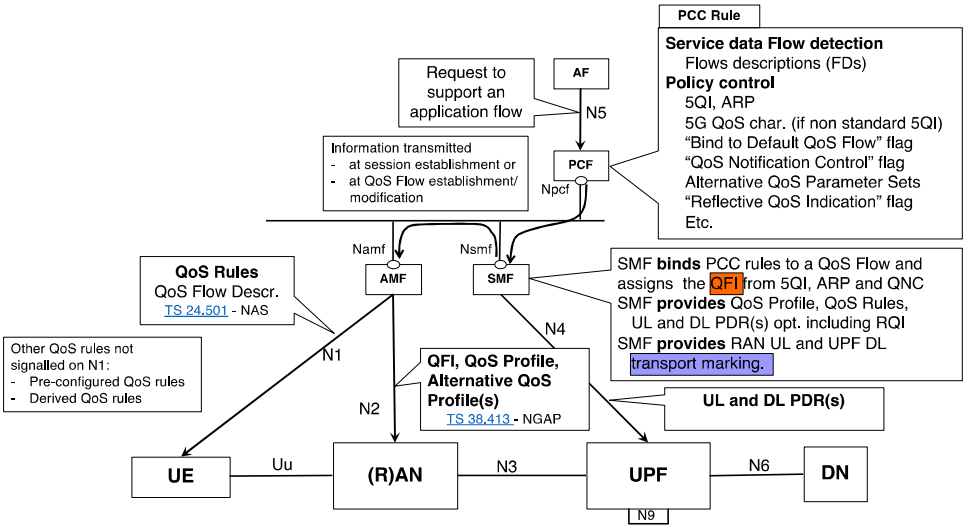


Figure 21 E2E QoS management

The Packet Filter Set is used by QoS rules (UL) and by PDRs (DL) to identify one or more packet flows. Based on the traffic header information such as source and destination IP address and port, Flow label for IP type PDUs, the packet filter will match this to find the appropriate flow. For Ethernet type PDUs the filter set is based on source/destination MAC address, Customer-VLAN tag (C-TAG) and/or Service-VLAN tag (S-TAG) VID or PCP/DEI fields as defined in IEEE 802.1Q [IEEE18-8021Q]. Figure 22 shows the usage of packet filters to map to QoS flows. The default QoS Flow always exist in a PDU session and is applied to a Non-GBR QoS Flow that do not match any of the filters.

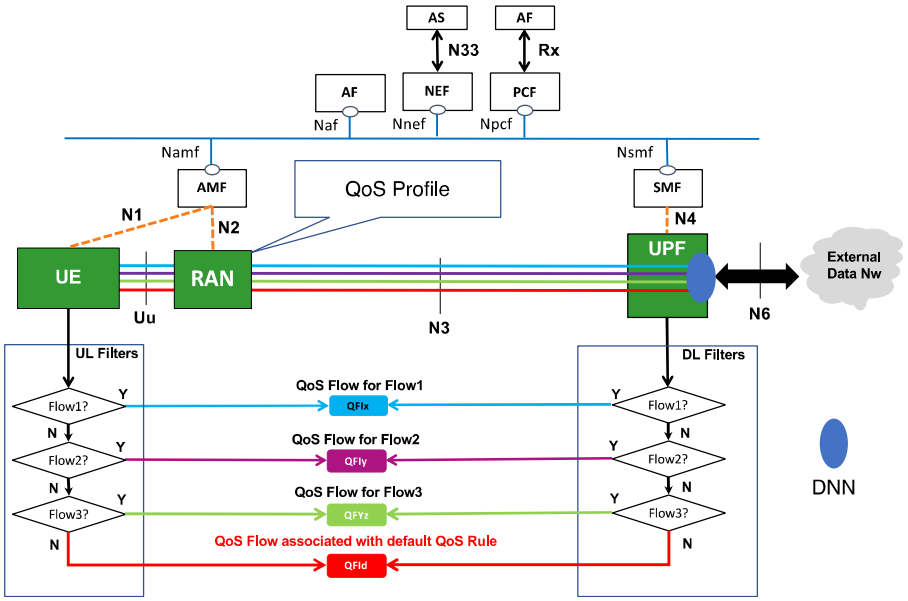


Figure 22 QoS differentiation by QoS flows

Using QoS differentiation in non-public networks

The QoS model described above applies to all the NPN deployment models. The main impact of NPN is on OAM (Operation and Management). The management of network configurations (such as QoS profiles, subscription data, etc.) affecting the QoS supported in a NPN is performed by the Network Operator. During the network planning and integration phases, the NPN user needs to clearly elaborate the different service flows needed by the applications and the performance objectives that are required to be met. The NPN operator will then dimension the network so that the necessary QoS performance objectives can be met. All of the above is captured in an SLA between the NPN user and NPN operator. The Network Operator can expose management service to the Industrial party/Third Party by providing access to some parameters/configurations. The exposure of management services to the Industrial party/Third Party by the NPN operator is regulated by an SLA between the parties. The details on the exposure via 5G-SMART management framework are provided in Deliverable D5.5 [5G21-D55].

QoS network planning and dimensioning

A prerequisite for providing any kind of QoS guarantees is proper network planning and dimensioning. The dimensioning is based (among other considerations) on the traffic model that is expected in the network, with some margins. Typically, one would dimension the network so that under normal operation (i.e., according to the expected traffic model) no network resource would be utilized to more than 70%, thus leaving some margin for unexpected traffic spikes. Via OSS, one can then monitor the actual resource utilization of the critical network elements (radio base stations, backhaul, etc.). If the monitoring reveals that resource utilization in some parts of the network is consistently exceeding 70%, it is a sign that resources need to be increased in that part. This could be because the traffic model was wrongly estimated or that it has evolved over time. For instance, if a radio cell in one part of the factory is consistently exceeding 70% utilization (e.g., in terms of Physical Resource Block

utilization), it indicates that it may be time to add resources in that part of the factory, e.g., by adding radio cells, or that other nearby cells need to be repositioned so that they pick up more traffic from that area. The key is to provide enough margin to be able to detect potential resource shortages before they actually affect the QoS.

3 Network architecture assessment and analysis

This section dives into three main assessments performed within the 5G-SMART project. These assessments augment the analysis performed in the predecessor Deliverable D5.2 and it provides a view on how the 5G NPN can operate in such multi stakeholder environment. A high-level system reliability analysis is shown which highlights the importance of 5G NPN component's reliability aspects. The section ends with analysis of NPN interworking with Edge computing aspects.

3.1 NPN operation model qualitative analysis

As observed in D5.2 [5GS20-D52], 5G NPNs play a key role in enabling critical IIoT applications in various vertical industries. Among other features, 5G NPNs enable novel operation models, where the roles and responsibilities for setting up and operating the network can be distributed among several stakeholders, i.e., among the public mobile network operators (MNOs), the industrial party who uses the 5G NPN services and 3rd parties. This results in many theoretically feasible operation models for 5G NPN, each with its own advantages and disadvantages. We investigate the resulting operation models and identify a set of nine promising models taking into account today's practical considerations. Additionally, we define a framework to qualitatively analyze the operation models and use it to evaluate and compare the identified operation models.

In the D5.2 Deliverable, deployment models are presented without discussing roles and responsibilities of various stakeholders in the operation of the 5G network. So far, only deployment models are described in 3GPP and in industrial fora such as NGMN [NGMN19-5GE2E] and 5G-ACIA [5GACIA19-5GAI]. Operation models are a way to take into account the roles of different stakeholders involved in operating an NPN [AR+19] [5GS20-D52] [3GPP20-28807]. This work leaps furthers into defining operation models. An operation model specifies the assignment of roles to the stakeholders. The stakeholders and roles used in this report represent a simplified but important subset of all stakeholders and roles in an ecosystem. Taking this into account this section considers the operation aspects of 5G networks and analyses various relevant scenarios. For this purpose, we consider the following stakeholders and roles.

Stakeholders

MNO is the stakeholder who owns and manages a public land mobile network (PLMN). In view of the new ecosystem made possible by the 5G technology, MNOs also engage in value creation in vertical domains.

Industrial Party is the stakeholder who requests NPN services for performing a (group of) industrial task.

3rd Party is the stakeholder who provides equipment and/or services for deployment and management of NPN and cannot be categorized as MNO or industrial party.

Roles

NPN owner is the role of owning the NPN infrastructure and includes both hardware and software components.

Spectrum owner is the role of having the right to transmit radio signals in a certain frequency band.

NPN integrator is the role of setting up the NPN according to a chosen architecture making it ready to use.

NPN operator is the role of operating and managing the NPN on a day-to-day basis. The NPN operator also offers NPN services, and as such the NPN service provider role can be assumed a sub-role of the NPN operator (as defined in [5GS20-D52]).

NPN User is the role who uses the services offered by the NPN for performing a group of industrial tasks.

We should note here that, for the sake of simplicity, the term ownership is used in a more generic sense, and we do not distinguish between direct or indirect ownership. For instance, in case of spectrum ownership we do not differentiate between the ownership as a licensee or through a leasing agreement with a licensee.

In principle, except for the role of NPN user, which is exclusively assigned to the Industrial party, all other roles can be taken by any of the three stakeholders. Therefore, if we put the role of NPN user aside, in theory 81 distinct operation models can be identified (i.e., 3 (NPN Owner) × 3 (Spectrum Owner) × 3 (NPN Integrator) × 3 (NPN Operator). Nevertheless, not all of those combinations would be meaningful and likely in practice. In fact, several factors such as business interests of stakeholders, or local regulations in different geographical areas and countries make certain operational models more attractive than others and therefore more likely to materialize. Taking into account common practices in ecosystems of today's mobile communication operation one can develop rules of thumb, which will help to identify major operation models. Two examples of such rules of thumb are:

- If an MNO is the Spectrum owner, it is likely that the MNO also takes the role of NPN operator.
- A stakeholder who is NPN owner is likely to take at least one more role (e.g., Spectrum Owner, NPN Integrator or NPN Operator).

Accordingly, we have identified nine major operation models—as depicted in Table 5—which we believe are more likely to be adopted by the industry. In operation model 1 (OM1), the industrial party takes all the responsibilities for the NPN operation, i.e., the industrial party implements and integrates an NPN, obtains the spectrum for it, operates it and uses the corresponding NPN services without directly involving any other stakeholder. At the other end of the scale, we have operation model 9, where an MNO takes all the roles, and the industrial player, as the NPN user, relies on the services provided by the MNO. In between, there are options as represented by operation models 2 to 8, where the roles are assigned to two or more stakeholders.

Table 5 Main NPN operation models

	Owner	Spectrum	Integration	Operation
OM1	Industrial Party	Industrial Party	Industrial Party	Industrial Party
OM2	Industrial Party	Industrial Party	MNO	MNO
OM3	Industrial party	Industrial Party	3rd Party	3rd Party
OM4	MNO	Industrial Party	MNO	MNO
OM5	3rd Party	Industrial Party	MNO	MNO
OM6	3rd Party	Industrial Party	3rd Party	3rd Party
OM7	3rd Party	MNO	MNO	MNO
OM8	3rd Party	MNO	3rd Party	MNO

OM9	MNO	MNO	MNO	MNO

3.1.1 Inter-relation between operation model and deployment model

In deriving the operation models in Table 5 Main NPN there are no assumptions regarding the deployment models. In practice, however, the deployment and operation models are intertwined. Specifically, the choice of a deployment model can have an impact on the feasibility of an operation model and the other way around. Table 5 illustrates the feasibility of all combinations of deployment models and operation models, where in total three distinct patterns can be observed. A key factor in the feasibility analysis of a combination is whether an MNO is the NPN owner or not. If not, then only NPN1 can be adopted, since sharing of resources with an MNO (NPN2-NPN4) does not make sense if those resources do not belong to the MNO. This is the case for seven out of nine operation models, i.e., models 1-3 and 5-8. In case of OM4, NPN1, NPN2 and NPN3 are feasible. NPN4 is not feasible in this case mainly because the industrial party is the spectrum owner, which prevents the integration of NPN into a PLMN with public spectrum.

Finally, the operation model 9 features the highest level of flexibility, when it comes to the combination with deployment models. The reason is that in this model, MNO assumes all the roles and accordingly will have the flexibility to decide on the level of integration between NPN and the public network.

In our analysis of the operation models and their interrelation with the deployment models (Table 5 and Table 6) so far, we have made the assumption, that each role can be exclusively assigned to only one stakeholder, which in the rest of this section we refer to as *dedicated operation model*. There are many situations, however, where it is desired or beneficial from technical, business, or operational perspective to share a role partially or fully between two stakeholders[29]. An operation model with a shared role in it is referred to as *shared operation model*. A prominent case for a shared operation model is where an industrial party—as the NPN user—would like to reduce the burden and complexity of the network operation and management on one hand, and on the other hand, have a certain level

Table 6 Feasibility of combinations between deployment and operation models (OM). Green: the combination is feasible. Red; the combination is not feasible

	NPN1	NPN2	NPN3	NPN4
OM 1				
OM 2				
OM 3				
OM 4				
OM 5				
OM 6				
OM 7				
OM 8				
OM9				

of visibility into the network management and handle simple management functions like end user activation and deactivation. In this case, it makes sense that the NPN operation role is shared between

[29] In theory a role can be shared with more than two stakeholders, but it is unlikely in practice due to operational complexity.

Industrial party and another stakeholder who takes NPN operation role, i.e., MNO or 3rd party. A similar sharing concept can also be applied to network ownership, where the network resources are shared between an MNO and an industrial party. Besides the roles of NPN owner and NPN operation, other roles, i.e., NPN spectrum owner, NPN integrator and NPN user, are not subject to sharing, mainly due to the nature of these roles.

There are also interrelations between NPN deployment models and possibilities for sharing of NPN operation roles. More specifically, not all NPN deployment models can be easily combined with shared operation models - Table 7 depicts the combinations. In NPN 1, where the NPN deployment is standalone and isolated from any MNO network, sharing of roles only make sense for the NPN operator role and between industrial party and a 3rd party who is tasked with the network operation and management. In this case, there will be little incentive for an MNO to take the responsibility of the NPN in a shared mode because the NPN is fully isolated from the PLMN. In NPN2 and NPN3 deployment models, where parts of the networking resources—i.e., RAN or Core—are shared between the industrial party and an MNO, it makes sense to also share the NPN ownership and NPN operation between the two stakeholders. Finally, in NPN4, where all resources are realized in a dedicated manner by an MNO, a shared operation model is only likely for NPN operation role, where network management and operation functions are partially shared between the two.

Table 7 Combination of deployment model and shared roles. Green: the combination is likely. Red: the communication is not likely

	NPN Owner	NPN Operator	
	Shared btw. MNO & Ind. P	Shared btw. MNO & Ind.P	Shared btw. 3rd P & Ind. P
NPN 1			
NPN 2			
NPN 3			
NPN 4			

3.1.2 Evaluation of the operation models

In this section we evaluate the operation models presented above. In our analysis we focus only on the dedicated operation models as presented in section 3.1, since the number of the shared operation models can be excessively large depending on which part of a role (out of many) might be shared between two stakeholders. In our evaluations, the focus is put on the industrial party as the NPN user. That is, we evaluate each operation model against the identified metrics as performance measures from the NPN user perspective. Our evaluations focus mainly on technical and operational aspects of NPNs, and we do not consider potential business models, which might be adopted by various stakeholders. The latter has been addressed in D1.3 [5G21-D13].

Metrics

NPN operation readiness A 5G NPN has different control and management functions compared to other industrial communication technologies and will require a high level of competence to operate the network. NPN operation readiness considers the level of complexity that a stakeholder needs to deal with for operating a given NPN, which can vary depending on the type of the operation model

considered where different stakeholders can assume different roles. Specifically, this metric indicates how well a stakeholder needs to prepare to take up the NPN operation role.

Service continuity demonstrates the capability of ensuring connectivity when NPN devices move out of the coverage of an individual NPN. This is for example applicable for IIoT applications that require end-to-end connectivity across multiple NPNs and also in the PLMN.

Privacy and security demonstrate how an operation model fulfils the typically stringent security and privacy requirements of IIoT applications, which are critical for protecting sensitive operational data and control over the network infrastructure, e.g., for the security software updates. In our analysis, we assess the privacy and security jointly as a single metrics.

Deployment flexibility indicates the ability to set up and customize the network based on the needs of the IIoT applications. These needs may evolve over time and hence this flexibility is required over the lifecycle of the system/service. There are several possibilities in which an NPN network can be (re)configured to achieve certain objectives.

Scalability measures the ability to scale (e.g., expand the network capacity and/or coverage) the NPN network based on changing requirements or introduction of new IIoT use cases.

The five metrics mentioned above mostly deal with the technical and operational aspects of the NPN operation models. There are obviously other—mostly regulatory and business related—metrics, which influence the choice of an operation model for IIoT scenarios, but we do not include these in our evaluations. For instance, some NPN users may have multiple sites spread across the globe and may desire to make common deployment and operational choices. Global applicability is the metrics that captures this capability. The choice of the operation model based on global applicability highly depends on the regulatory aspects such as the availability of the spectrum options in selected countries of NPN deployment. A detailed investigation of spectrum licensing options and their relations to the NPN operation models is a very broad topic and goes beyond the scope of this work. For example, an industrial party can only be a spectrum owner if it can be a licensee or a lessee of the spectrum. If the regulation allows, it is possible to utilize both private industry spectrum and public spectrum for achieving specific purposes[30]. For instance, one can combine private and public spectrum to increase the available bandwidth. Also, it is possible that an MNO can offer an NPN solution operating in private spectrum instead of MNO licensed public spectrum.

Another crucial metrics is the total cost of operating NPN. While several of the above metrics have an impact on costs, a complete evaluation of the costs often requires information on commercial offers, which are not publicly available.

3.1.3 Analysis of operation model

For qualitatively evaluating the operation models against the metrics defined above three qualifiers are used: *High, Medium, and Low*. *High* means that the corresponding metrics is very well supported under the considered NPN operation model. *Low* is used to indicate that the metrics is either not supported or supported only with significant adaptations. Finally, *Medium* is used as a level between *High* and *Low*. As we will see below, the evaluation of operation models will in many cases depend

[30] In this case there are two separate Spectrum Owners, the role is not shared.

also on the adopted deployment architecture. Accordingly, presents three sets of evaluation results: the results for the case that NPN1 is adopted (applicable to all operation models), the results for the cases that NPN2 or NPN3 are adopted (applicable to OM4 and OM9), and finally those for scenarios with NPN4 (applicable only to OM9).

In evaluating the NPN operation readiness, we argue that it only depends on who operates an NPN. If an NPN is operated directly by an Industrial party (i.e., OM1), then the initial operation readiness would be Low, since the operation of a cellular network requires a certain level of competence and know-how, which is usually not available within an OT enterprise and needs to be build up. In contrast, the operation readiness level is evaluated as High in all other cases (OM2-OM9) since the operation is done by an experienced MNO or 3rd party.

Similar to the NPN operation readiness, the service continuity of operation models also depends on who operates the NPN in question. Here, if the NPN operator is not an MNO (i.e., OM1, OM3 and OM6), then the service continuity will be considered low, since it requires a separate agreement between the NPN operator and an MNO, which needs to be negotiated. On the other hand, for other models, where an MNO operates the NPN, the service continuity can be evaluated between Medium to High depending on the adopted deployment model. Specifically, offering service continuity would be much easier (High) if the NPN is highly or fully integrated in the PLMN (NPN 4 and 3) and it will be less straight forward (Medium) if the NPN is stand-alone (NPN 1) or only partially integrated into the PLMN.

The privacy and security of operation models depend on who will act as the NPN owner and NPN operator, since these might have access to the user data and metadata (e.g., NPN control and management data). There are established methods that can be applied to ensure the requested security and privacy of the user data and metadata in all the operation models in Table 5. Examples of such methods include application-level encryption, service level agreements (SLA) and network slicing [5GACIA19-SA]. Having said that, the security and privacy metrics for various operation models can still be evaluated and ranked based on a) the potential visibility of metadata to actors other than the Industrial party and b) the degree of control that the Industrial party is able to exercise on operations. Accordingly, we evaluate the security and privacy as High for the OM1, where only the industrial party has visibility to the metadata, and it has full control over the adopted security and privacy measures. All other operation models (OM2-OM9) are evaluated as Medium or Low in Table 8 depending on if one or two other stakeholders (besides the industrial party) are involved in ownership and operation of the NPN, respectively. Obviously, a higher number of stakeholders might increase the potential security and privacy risks (the so-called attack surface) and make the control over security measures more complicated.

The deployment flexibility depends on the allocation of NPN owner and operator roles. In particular, the industrial party will enjoy the highest level of flexibility in customizing the NPN deployment if both the NPN owner and operator roles are assigned to it (OM1). In other cases, the industrial party will need to rely—at least partially—on 3rd party or MNO capabilities for NPN customizations. In these cases, if the industrial party is still the owner (OM2 and OM3), we evaluate it as Medium, since it will still have good leverage to customize the NPN. In other cases, i.e., OM4, OM5 and OM7-OM9, the flexibility is considered Low. The only exception here is the case, where a 3rd party is both the NPN

owner and operator. This case we evaluate also as Medium, since a 3rd party might provide comparatively a better support for customization than an MNO.

For the scalability of the operation models, the adopted NPN deployment model plays a key role. Specifically, the more an NPN is integrated into the PLMN, the higher its scalability will be, because MNOs usually have great amounts of resources at hand to scale up a PNI-NPN if needed. Accordingly, we evaluate the scalability of operation models adopting NPN1 with Low, those adopting NPN2-NPN3 with Medium and the ones adopting NPN4 with High.

Another important aspect of the NPN operation models analysis, which is not directly obvious from Table 8, is the impact of direct spectrum allocation to the industrial party (the so-called local spectrum). From an industrial party's point of view the interest in local spectrum could be due to independence from an MNO (i.e., OM1 will not be possible without the local spectrum) and/or having dedicated spectrum for the IIoT use cases. In practice, an MNO has all the means to fulfil the latter requirement, e.g., through adopting network slicing. Therefore, once the RAN equipment supports the frequency bands used for local spectrum, the functioning of the system is not affected by the fact that local spectrum is used.

Summarizing the analysis of the operation models, we observe that there is no single operation model that optimizes all the metrics. It also illustrates that an NPN user has to make trade-offs when choosing the right operation model. As an example, Table 8 Result of the operation models analysis. For the result indicated with a, b and c the assumption is that deployment architectures NPN1, NPN2/NPN3 and NPN4 are adopted, respectively. If not indicated explicitly, the results apply to all NPN1-NPN43 demonstrates a clear trade-off between NPN operation readiness on one hand and privacy and security as well as deployment flexibility on the other hand, such that not all these metrics can be optimized to High at the same time. Therefore, the right operation model varies depending on how different metrics are prioritized for a certain smart manufacturing scenario. For instance, assuming the NPN1 deployment architecture, an NPN user might choose:

- OM1 to have High for security and privacy,
- OM2, OM3 or OM6 to get at least Medium for security and privacy, as well as for operation readiness and deployment flexibility.

Table 8 Result of the operation models analysis. For the result indicated with a, b and c the assumption is that deployment architectures NPN1, NPN2/NPN3 and NPN4 are adopted, respectively. If not indicated explicitly, the results apply to all NPN1-NPN43

	Operation simplicity	**Service continuity**	**Privacy & Security**	**Deployment flexibility**	**Scalability**
OM1	Low	Low	High	High	Low
OM2	High	Medium	Medium	Medium	Low
OM3	High	Low	Medium	Medium	Low
OM4	High	Medium	Low/Medium	Low	Low[a] (Medium[b])
OM5	High	Medium	Medium	Low	Low
OM6	High	Low	Medium	Medium	Low
OM7	High	Medium	Medium	Low	Low
OM8	High	Medium	Medium	Low	Low
OM9	High	Medium[a, b] (High[c])	Low/Medium	Low	Low[a] (Medium[b]/High[c])

3.1.4 Summary

We identified and presented nine most plausible operation models of 5G NPN for smart manufacturing scenarios and elaborated on how these models differ from each other depending on how the roles and responsibilities are distributed among different stakeholders. Additionally, we have proposed a framework for a systematic analysis of NPN operation models, including a set of metrics from an NPN user perspective to qualitatively analyse and compare the models. Our analysis highlights the trade-offs that are involved in the selection of the right NPN operation model. At the end, each NPN user has to apply its own weights and priorities in order to derive preferable options. While in our analysis we mainly looked at the technical and operational aspects of the NPN operation, the final choice of the NPN operation model will depend also on other regulatory- and business-related metrics such as spectrum allocation policies, cost, liability protection, lock-in protection, and business models.

3.2 Network reliability analysis

Introduction to some basic reliability engineering concepts

Reliability engineering deals with the longevity and dependability of a complex system and its subcomponents. Systems may fail (not provide the service they are designed for) and complex systems will fail in non-predictable ways. Hence the analysis of reliability is inherently stochastic in nature. This also applies to 5GS being a highly complex system.

Reliability is defined as the probability that an item will perform a required function without failure under stated conditions for a given period of time [3GPP21-22104].

Reliability is quantified in terms of the mean number of failures in a given time (failure rate), or as the mean time between failures (MTBF) for items which are repaired and returned to use (in our analysis we consider only repairable systems). In most basic analyses constant failure rate (λ) is assumed and in this case $MTBF = 1 / \lambda$.

For repairable systems, the other important term is availability. Availability is the probability that a system is available at a given time (t) provided that it was working at an earlier time $(t0)$.

Under steady state conditions, the relation between availability and reliability is provided below, where MTTR is the mean time to repair a system:

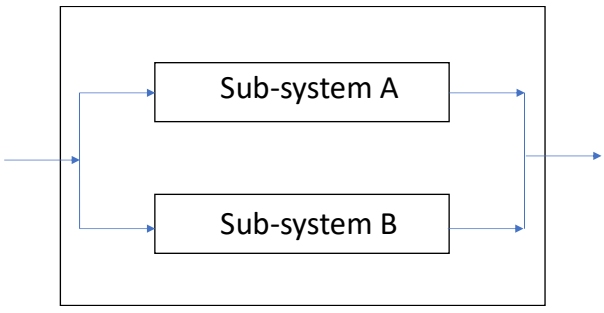
$$Availability = Uptime / (Uptime + Downtime) = MTBF / (MTBF + MTTR).$$

It should be kept in mind that MTTR is a critical element in providing high availability. As a concrete example if a system has on the average one failure a year and the repair time is 24 hours, the availability is 0.997 (less than 3 nines). If the repair time can be brought down to 1 hour, the availability goes up to 0.99988 (close to 4 nines).

When we consider E2E communication in 5GS, there are several sub-systems (each with their own availability characteristics) that are linked together to provide finally an E2E availability. The manner in which they are linked – whether sub-systems are in series or in parallel has a major impact on the E2E availability.

E2E availability for a system whose sub-systems are connected in series: $Ae2e = A1 * A2 * \dots * A_n$, where An is the availability of the ith sub-system. To see the implications let us assume a system made of 5 sub-sytems each of which has a 99% availability. The E2E availability is only 95% as seen by the following - $Ae2e = 0.99 * 0.99 * 0.99 * 0.99 * 0.99 = 0.95$. Another example with different availabilities – A_1, A_2, A_3 are all 0.9 and A_4 is 0.1. Then $Ae2e = 0.9 * 0.9 * 0.9 * 0.1 = 0.0729$

The main observation that can be concluded from the above discussion is that in a system configured in series fashion, the E2E availability goes down with the length of the chain and the E2E availability is lower than that of the weakest link. Now let us suppose that we can make the system completely parallel. A simplified view of an E2E parallel system is shown below (Figure 23).



There are two disjoint paths through the system. Hence failures in one path do not affect the other path. If the unavailability of each path is 0.01 (availability 0.99), the unavailability of the system is 0.01 * 0.01 = 0.0001. Hence the availability of the system is 1 - 0.0001 = 0.9999. From 2 nines at sub-system level to 4 nines at system level.

Figure 23 System configured in parallel fashion

Formally the availability of a system with N parallel components can be calculated as $A_{e2e} = 1 - \prod_{i=1}^{N}(1 - A_i)$, where A_i is the availability of individual component.

5G architecture for industry and reliability

This section will apply the notions introduced above to analyse the implications on architecture for 5G deployments in the context of industrial applications. It should be kept in mind that the description below remains high level for the following reasons:

- The sub-systems involved in 5G NPN deployment are complex involving hardware, power supply and other physical systems such as cables and antennas in addition to multiple software layers (virtualization framework, operating system, network functions and so on).
- Equipment manufacturers' data on availability is confidential between NPN operator and NPN vendor
- The wireless communication part is covered in deliverable D1.4 [5GS20-D14] and we use a simplified approach based on URLLC objectives for the provided analysis.
- 5G-SMART D1.4 has already investigated extensively the reliability aspects of the radio connectivity link for various NPN deployment options]. The current analysis focuses more on the deployment reliability aspect of the 5GS NPN (e.g., system reliability).

Figure 24 illustrates the architecture used for the analyses in this section. An end device connects to the 5G system through a UE. The UE uses the RAN to the send the data in a wireless manner to a gNodeB (base station). The data is then sent on the UPF and then onto a DN (Data Network). This analysis focuses on the user plane, once a PDU Session has been opened, as it can be naturally understood that reliability is primarily a user plane issue. We study the contribution of the 5G system as shown by green dotted line in Figure 24.

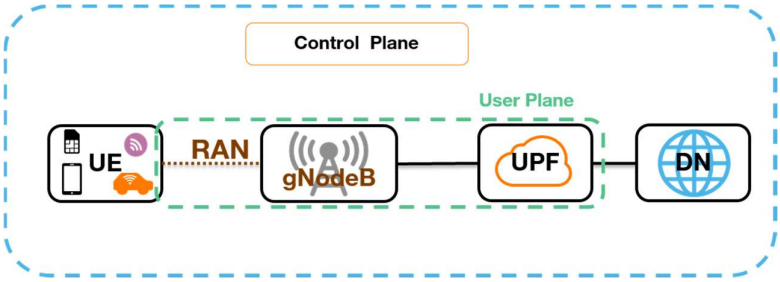


Figure 24 5GS User plane Architecture

Assuming that the sub-systems are in series, schematically we have 4 subsystems in our E2E chain. This is shown as a simple Reliability Block Diagram (RBD) below (Figure 29). It should be noted that the UE and the data network are not included in the analysis provided in this section as we focus on the contribution of the 5G system RAN, transport and core components. Another manner to state this is to consider the availability of the UE and the DN as 1.

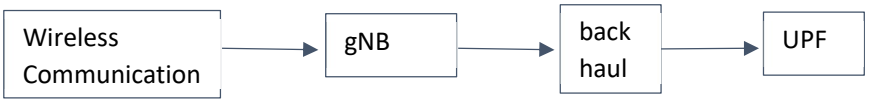


Figure 25 Reliability block diagram of the E2E chain

From Release 15 onwards, 3GPP has defined a number of features to support very low latency combined with high reliability (URLLC). These include features such as flexible sub-carrier spacing, a sub-slot-based transmission scheme, new channel quality indicator, new modulation and coding scheme tables [3GPP21-38331]. For the Wireless Communication sub-system, the availability design objective for URLLC in Release 15 is 99.999 % (five nines) which is considered here as the baseline.

For the backhaul it is assumed a fibre link over short distances (<1km) so that no intermediate switches are needed. For such short distances on fibre the availability can be assumed to be 1 (i.e., perfect link).

The UPF is assumed to be on a virtualized host. This is a system comprising of power supply, hardware, operating system and a virtualization framework. In order to obtain figures for availability for UPF one can examine the SLAs for major IaaS/cloud service providers. The actual availability is highly dependent upon the agreed SLA between service providers and consumer. However, based on the currently available platforms, 4 nines 99.99 % of availability may be considered as the upper end for single Virtual Machine SLAs. We will use this IaaS figure of 99.99 % availability in our analyses.

The gNodeB is in reality a complex system by itself. A gNodeB may have a split between a baseband unit (BBU) and a remote radio unit (RRU). The gNodeB could make use of a distributed antenna system (DAS). A detailed analysis of gNodeB availability is not in the scope of this deliverable. We rely on figures provided by the equipment manufacturer. However, the performance characteristics are not publicly available. So, in our analysis we decided to use the IaaS figures as a baseline and apply a penalty for the antenna system which is not present in an IaaS. Hence, for the gNodeB we will use 99.95 % as the availability figure.

Let us take a URLLC type use case that needs an E2E availability of 5 nines. Using the series model the E2E availability of the above system is $Ae2e = ARAN * AgNodeB * AUPF * AUPF = 0.99999 * 0.9995 * 1 * 0.9999 = 0.9994$ which is less than 4 nines. In order to improve the availability of the weakest link, which is the gNodeB (in the example we are using for illustration), an equipment supplier may propose a high availability version of their gNodeB. This in reality includes redundancy (and hence parallelism) to the product by adding redundant power supplies, disk arrays (parallelising storage) and other critical components. Even if this improves A_{gnodeB} to 5 nines (99.999%), the E2E availability will still be less than 5 nines (99.999%).

Another approach is to utilise redundancy to provide disjoint paths from UE to UPF. As discussed in section 3.4.1 and shown in Figure 26we use the dual connectivity redundancy provided by the 5G system.

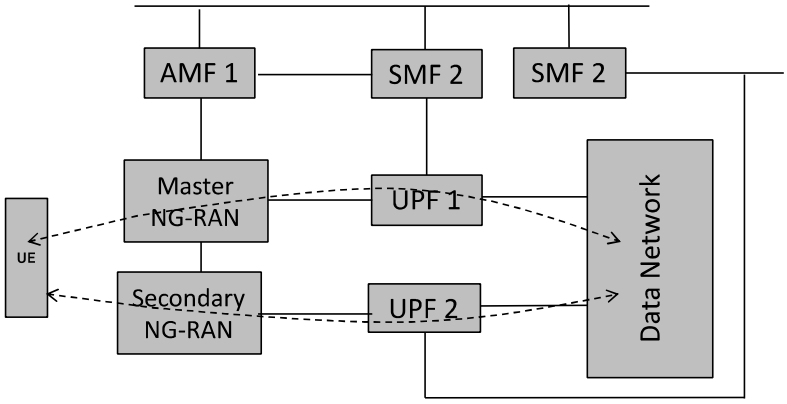


Figure 26 Redundancy in E2E chain

This will provide us with a parallel system with two independent paths where each individual path has the availability calculated above. Let us assume that the availability of each sub-system in the path is the same. In the parallel model the $A_{e2e} = 1 – ((1 – 0.9994) * (1 – 0.9994)) = 0.9999996$ which is better than our objective of 5 nines.

Lessons learnt from above example

- A realistic evaluation of E2E availability necessitates knowledge of the deployed architecture and access to the equipment manufacturers' data for the availability of all the sub-systems.
- Over the entire E2E chain, it is important to examine which subsystems are in series and those that are in parallel so that a Reliability Block Diagram may be drawn.
- Once the precise deployment architecture choices are known, calculation of the E2E availability is straightforward with some knowledge of reliability engineering.
- In order to achieve 5 nines E2E availability, redundancy is a must. For example, connectivity to more than one gNBs can provide significant gains in reliability to achieve the E2E objective
- The mean time to repair (MTTR) is a critical component of availability. To the extent possible, systems should be self-healing or allow online detection and repair in order to minimize downtime.
- In addition to ensure high reliability of the radio link with advanced URLLC feature, it is of similar importance to ensure reliability of the set of physical and virtual components in 5GS E2E chain (e.g., gNodeB, UPF).

3.3 NPN interworking with edge computing

The integration of various NPN deployment options with edge computing capability shown in Section 2.3 enables high flexibility for the OT players to find the solution, which meets their service requirements. The goal of this section is to analyze and evaluate the main characteristics of the different integration options.

The on-premise, private edge, together with the standalone NPN provides a lots of freedom for the enterprise customers in the deployment customization in order to fulfill the specific industry requirements. The edge owner or a 3rd party integrator can characterize the datacenter hardware and software portfolio as well as configure and manage the edge deployment according to the specific

application requirements of the industry party. GPUs can be deployed to support the compute intensive tasks (e.g., video feed processing, AI acceleration). Smart NICs can also be applied to support load balancing, path optimization and these enable to offload networking related functions (e.g., virtualized TSN FRER functionality) from the server to the NIC. Furthermore, the datacenter hardware infrastructure and networking can be designed for ensuring high reliability (e.g., enabling the deployment of multiple active application instances using completely separated infrastructure resources). The factory local edge also provides that the sensitive data (e.g., device control application data) is kept within the factory premise.

Since the standalone NPN and the edge is handled by the same owner/integrator this option provides the tightest interworking between the different domains (legacy Industrial LAN/TSN domain, 5G domain, edge computing domain), resulting an integrated end-to-end solution by leveraging the NPN and edge capabilities e.g., for providing ultra-low latency. Since specific, virtualized TSN functions (e.g., FRER) can be deployed in the edge datacenter, the seamless interworking between the TSN and the cloud domains can also be established, by supporting e.g., the integration of TSN FRER and cloud-based reliability solutions for end-to-end reliability (described in section 2.4.3), as well as the supporting of TSN scheduled traffic feature by real-time capabilities in the cloud domain.

When the cloud infrastructure consists of cooperating Kubernetes central and edge clusters, then the deployment must decide on whether distributed or centralized Kubernetes control-plane (that manages the worker nodes and the Pods) approach is the more suitable. The centralized control-plane approach induces the risk of separation of an edge cluster from the central cloud as it is not possible to provision nor to reconfigure workloads hosted on unreachable nodes at the edge. While the distributed control-plane approach can manage workloads on each site even in the case of disconnected network between the edge and central sites.

Considering Kubernetes, as the container orchestration platform: Currently the standard Kubernetes distribution needs to be customized to some extent to fulfill the requirements dictated by demanding industrial applications. Many aspects involve the configuration and setup of the hardware resources that are hosting the cloud platform, in this case, Kubernetes. They can be adjusted if, the hardware is in the supervision of the owner/integrator, and then custom options can be set, such as providing real-time support at the operating system level of the datacenter servers.

At Kubernetes level configuration it is recommended to define no CPU limits or to disable the enforcing of CPU limits to support low latency workloads. In addition to that, Kubernetes tools can be used that support better performance isolation for selected pods to serve sensitive workloads, by partitioning the Kubernetes nodes and run such workloads only on nodes tuned for low latency.

However, as Kubernetes is highly customizable, this is viable, but still, common industrial extensions, add-ons that provide platform level solution for the problems could drive the usage of Kubernetes better in industrial edge computing environments. For example, to launch multiple application instances with intrinsic data replication between them either in hot standby or active-active resiliency mode and provide a single service to hide them from the device side, could be a desirable service construction object in industrial environments.

It also must be noted that when a service is deployed in an edge computing environment then the configuration of the edge features including the internal networking and the networking (5G connectivity) between the edge computing platform and the devices has to be done together.

In the implementation of many 5G-SMART use cases (UC) [5GS20-D11] the factory-local edge solution was applied. For example, UC1 5G-Connected Robot and Remotely Supported Collaboration, UC2 Machine Vision Assisted Real-time Human-Robot Interaction over 5G [5GS20-D31]and UC 6, Cloud-based mobile robotics [5GS21-D21] trials demonstrated that the local edge solution fits for the low-latency requirements of mobile robot control. In UC6 the control software components run in containers within the Kubernetes platform deployed on the factory-local servers. There is also an example where a virtual machine is needed in the cloud, in UC 4 5G for Wireless acoustic workpiece monitoring, the monitoring software component Genior Modular (GEM) is deployed on a virtual machine, as it is based on a heavily customized operating system image [5GS21-D32].

On the other hand, if the on-premise edge is deployed, configured, managed and operated by the industrial party, it requires extra skills and knowledge in the cloud and networking domains. As shown in Section 2.3.4, several on-premise edge deployment options are possible, but the management of all these options is not yet fully supported by consumer Kubernetes toolset, so specific custom resources and configurations have to be applied. Alternatively, an integrator (3rd party integrator, NPN vendor, etc.) can also handle the deployment, integration, and management tasks.

The PNI-NPN with shared RAN and core control plane option enables the mobile network operator to be more involved in the enterprise/industrial deployment. From the enterprise customer perspective this option could be a good equilibrium, since the (sensitive) user plane traffic still remains on-premise enabling the support of low-latency, and even deterministic communication in a secure way, but some control-plane tasks - including the edge computing-related management tasks - are handled by the MNO. From edge computing perspective it means that the MNO can provide a PaaS solution for the enterprise customer, who can then concentrate on the handling of the industry applications. Albeit a PaaS solution is offered, but the edge infrastructure is still deployed on the factory premise, so the infrastructure capabilities can be adjusted to the specific requirements of the industrial applications.

The option when the PNI-NPN is hosted by the public network could be suitable for such Industry use cases, when the service requirements are not so strict, and it is not critical if the data goes outside the factory premise. Furthermore, this scenario is suitable for such use cases, where service continuity is important: when the UE moves to a new location, and different edge server is selected, the minimization of the service interruption is a crucial point for industry applications. If the footprint of the MNO enables to deploy edge computing resources close enough to the factory premise, then applications with low-latency requirements can also be supported. In PNI-NPN deployment both scalability and reliability can efficiently be ensured on the network side, and if the related edge computing solution (deployed on a factory premise or provided by the MNO as a PaaS) can ensure these to the same extent, then the end-to-end solution can utilize these advantages. As it is discussed in Section 2.3.5 many business and technical alternatives are possible, when the MNO and 3rd party providers could offer IaaS or PaaS for the enterprise customers – however, the details of business relationships are not clear yet and there are also some technical issues with this scenario. As an advantage, a customized cloud management (e.g., Kubernetes) service is offered, so the industrial

party can concentrate on the handling of its applications. On the contrary, a PaaS solution enables less flexibility than an on-premise, private edge, so probably a limited set of industrial applications can be supported. For example, the edge infrastructure (e.g., hardware resource) cannot be reached directly, which causes difficulties, e.g., for configuring a seamless TSN FRER – edge reliability integration. Furthermore, albeit edge computing related standardization work is on-going in 3GPP SA2, SA5 and SA6 and a set of solution are available there are still open questions, regarding e.g., edge application discovery or edge computing management.


5G
smart

4 Conclusion

5G NPN deployment models are set to become prominent network architecture options for the 5GS integration with smart manufacturing eco-system. From E2E perspective, there are several aspects important in order to ensure the functional requirements of the smart manufacturing applications. This report provides an in-depth investigation from device to Edge cloud integration aspect for different 5G NPN deployment models. The report takes a further leap by extending the network architecture concepts introduced in 5G-SMART Deliverable D5.2 and completes the network architecture analysis of operation models and network reliability analysis.

Starting with device architecture aspects, it is observed from analysis that a unique device architecture will not fulfil the complete wide range of communication characteristics for all the applications. The report highlights three different themes based on the communication characteristics of smart manufacturing applications. Considering importance of the 5G integration with Ethernet-based industrial networks such as TSN, a reference device architecture that incorporates interaction with Ethernet endpoints (Ethernet bridge/end-station) is proposed.

Key resilience enablers are highlighted, and mechanisms supported by 5GS providing support for resiliency are elaborated. Amongst the diverse resiliency enablers, this document focuses on redundancy which is a key enabler for high availability. It can be observed that with the current 5GS mechanisms, high availability smart manufacturing communication services can be supported.

Network reliability analysis shows all the relevant physical and virtual components involved in 5G NPN operations. It can be deduced that, along with ensuring reliability of 5G connectivity, one needs to ensure high availability of the hardware and software components involved in E2E NPN architecture. Further, analysis shows the necessity of having parallel redundancy of the components in order to fulfil the demanding requirement of the smart manufacturing applications.

An extensive analysis of different Edge cloud integration options with NPN deployment models is performed. Several learnings are listed from this analysis. Depending upon the smart manufacturing use cases, cloud deployment options from standalone Edge data center, federation of Edge data center, integrated Edge and central cloud premises can be selected.

5G system with URLLC features, TSN support and integration with Edge cloud can provide E2E deterministic communication services for wide range of smart manufacturing use cases. The present report highlights the main open challenges of adding support for TSN features such as FRER in an Edge cloud platform which interworks with the 5GS.

Following this, several plausible solutions for integration of TSN with 5G enabled Edge cloud systems are proposed. These solutions are categorized based on the device capability. These capabilities include the ability to handle multiple application instances as opposed to a single application instance. New Interworking function is proposed which enables TSN FRER functionality within virtualized environment. The analysis shows that the Kubernetes as an orchestration platform can provide such flexibility for a wide range of the Edge cloud deployments with NPN. In addition, Kubernetes as an Edge computing platform has a capability to ensure containerized application being well integrated with 5G-TSN industrial networks.

A set of nine plausible NPN operation models are proposed. These models provide an understanding on how the roles involved in operating a 5G NPN deployment can be assigned to different stakeholders in the smart manufacturing eco-system. To evaluate such NPN operation models, a framework is proposed for systematic analysis of different NPN operation models. A set of performance measures from NPN user perspective is provided. NPN users can utilize such framework by applying their own weights and priorities for different performance measures in order to derive preferable options, leading to the selection of NPN operation model suitable to the NPN user needs. The analysis is limited to operational and technical aspect of the NPN operation. The final choice of NPN can further depend also upon regulatory and business-related measures such as spectrum allocation policies, cost, liability protection, lock-in protection, and business models.

Reference

[5GS20-D52]	5G-SMART deliverable D5.2, "First report on 5G network architecture options and assessments", November 2020. https://5gsmart.eu/wp-content/uploads/5G-SMART-D5.2-v1.0.pdf
[5GS20-D11]	5G-SMART deliverable D1.1, "Forward looking smart manufacturing use cases, requirements and KPIs", June 2020. https://5gsmart.eu/wp-content/uploads/5G-SMART-D1.1.pdf
[3GPP21-22104]	3GPP Technical Specification TS 22.104,"Service Requirement for cyber-physical control applications in vertical domain" September, 2021
[3GPP20-23501]	3GPP technical specification TS 23.501, "System architecture for the 5G System (5GS)", Release 16, August, 2020
[IETF-EAP78]	IETF RFC 3748, "Extensible Authentication Protocol (EAP)", 2004, https://datatracker.ietf.org/doc/html/rfc3748
[ResiliNets]	www.resilinets.org
[3GPP21-23527]	3GPP technical specifcication TS 23.527, "5G system; Restoration procedures" September 2021
[3GPP21-23548]	3GPP technical specification TS 23.548, "5G System Enhancements for Edge Computing; Stage 2" September 2021
[3GPP21-23558]	3GPP technical specification TS 23.558, "Architecture for Edge Applications", September 2021.
[3GPP21-28814]	3GPP technical report TR 28.814, "Management and orchestration; Study on enhancements of Edge computing management", September, 2021
[ETSI-MEC20]	ETSI, "Harmonizing standards for edge computing - A synergized architecture leveraging ETSI ISG MEC and 3GPP specifications", ETSI White paper, 2020
[GSMA20-TEC]	GSMA, "Telco Edge Cloud: Edge Service Description and Commercial Principles", October 2020, GSMA white paper, https://www.gsma.com/futurenetworks/resources/telco-edge-cloud-october-2020-download/
[IEEE17-8021CB]	IEEE Standard for Local and Metropolitan Area Networks- Frame Repliciation and Elimination for Reliability
[IEEE18-8021QCC]	IEEE, "P802.1Qcc Draft Standard for Local and metropolitan area network - Bridges and Bridged Networks - Amendment:Stream Reservation Protocol (SRP) Enhancements and Performance Improvements," IEEE, 2018.
[5GS20-D51]	5G-SMART deliverable D5.1, "First report on new technological feature to be supported by 5G standardization", June 2020.
[NGMN19-5GE2E]	Dhruvin Patel, Joachim Sachs, NGMN, "5G E2E technology to support verticals URLLC requirement", October 2019.
[IEEE18-8021Q]	IEEE, "IEEE standard for Local and Metropolitan Are Networks - Bridges and Bridged Networks IEEE Std 802.1Q-2018," IEEE, 2018.
[5GACIA19-5GAI]	"5G for automation in industry," 5G-ACIA Whitepaper, March 2019. [Online]. Available: https://www.5g-acia.org/index.php?id=6960
[AR+19]	A. Rostami, "Private 5G Networks for Vertical Industries: Deployment and Operation Models," 2019 IEEE 2nd 5G World Forum (5GWF), Dresden, Germany, 2019, pp. 433-439, doi: 10.1109/5GWF.2019.8911687.

[5GS21-D13] 5G-SMART deliverable D1.3, "Operator business models for smart manufacturing", June 2021, https://5gsmart.eu/wp-content/uploads/5G-SMART-D1.3-v1.0.pdf

[3GPP20-28807] 3GPP technical report TR 28.807, "Study on management of Non-Public Networks (NPN)," Release 16, 2020.

[5GS20-D14] 5G-SMART deliverable D1.4, "Radio network deployment options for smart manufacturing", November 2020, D1.4 (5gsmart.eu)

[5GACIA19-SA] Security Aspects of 5G for Industrial Networks," 5G-ACIA Whitepaper. [Online]. Available: Security Aspects of 5G for Industrial Networks - 5G-ACIA (5g-acia.org)

[5GS21-D21] 5G-SMART deliverable D2.1, "Design of 5G-based Testbed for industrial robotics, et. al", November 2020, https://5gsmart.eu/wp-content/uploads/5G-SMART-D2.1.pdf

[5GS21-D32] 5G-SMART deliverable D3.2, "Report on system design options for monitoring of workpieces and machines", November 2020, https://5gsmart.eu/wp-content/uploads/5G-SMART-D3.2.pdf

[3GPP21-38331] 3GPP technical specification, "NR; Radio Resource Control (RRC); protocol specification", September 2021

[5G21-D55] 5G-SMART deliverable D5.5, "Report describing the framework for 5G system and network management functions", November 2021, https://5gsmart.eu/wp-content/uploads/5G-SMART-D5.5-v1.0.pdf

List of abbreviations

3GPP	3rd Generation Partnership Project
AMF	Access and Mobility Management
AP-BBIF	Application Processor-Baseband processor interface
BBU	Baseband Unit
CFS	Completely Fair Scheduler
CNC	Centralized Network Controller
CNCF	Cloud Native Computing Foundation
CNI	Container Network Interface
CUC	Centralized User Controller
DC GW	Datacenter gateway
DS-TT	Device Side TSN Translator
DS-TT	Device Side TSN Translator
E2E	End to End
EAP	Extensible Authentication Protocol
EAP	Extensible Authentication Protocol
EAP	Extensible Authentication Protocol
eUICC	Embedded Universal industrial circuit card
FRER	Frame Elimination and Replication
GBR	Guaranteed Bit Rate

gPTP	Generalized Precision Time Protocol
IIoT	Industrial internet of things
ISP	Image Signal Processor
LCM	Life-Cycle Management
MDBV	Maximum Data Burst Volume
MNO	Mobile Network Operator
MOCN	Multi-Operator Core Network
MORAN	Multi-Operator RAN
NAS	Non Access Stratum
NIC	Network Interface Card
NPN	Non-Public Network
OAM	Operations Administration and Maintenance
OAM	Operation and Management
Paas	Platform as a Service (PaaS)
PCF	Point Coordination Function
PDR	Packet data rules
PDR	Packet Data Rules
PDU	Packet Data Unit Session
PFD	Packet Flow Description
PMIC	Port management information container
PMIC	Port management information container
PMU	Power and clock management
PN	Public Network
PSA	PDU session anchor
QFI	QoS Flow ID
RQA	Reflective QoS
RRU	Remote Radio Unit
SaaS	Software as a service
SBI	Service Based Interfaces
SDF	Service Data Flow
SDF	Service Data Flow
SLA	Service Level Agreement
SMF	Session Management Function
SNPN	Standalone Non-Public Network
TS	Time Sync
TSN-BB IF	TSN Baseband Interface
TSN-IWF	TSN Interworking Function
UDSF	Unstructured data storage function
UDSF	Unstructured Data Storage Function
UICC	Universal Integrated Circuit Card

UICC	Universal industrial Circuit Card
UPF	User Plane Function
VM	Virtual Machine
VM	Virtual Machine

Table 9: List of abbreviations