ZEROSWARM

D3.4 - 5G edge computing devices & services

ZERO-enabling Smart networked control framework for Agile cyber physical production systems of systems

Project funded by Horizon Europe, Grant Agreement #101057083

Topic	HORIZON-CL4-2021-TWIN-TRANSITION-01-08
Project Title	ZERO-enabling Smart networked control framework for Agile cyber physical production systems of systems
Project Number	101057083
Project Acronym	Zero-SWARM
Contractual Delivery Date	M17
Actual Delivery Date	M17
Contributing WP	WP3
Project Start Date	01/06/2022
Project Duration	30 Months
Dissemination Level	Public/Sensitive
Editor	SICK AG
Contributors	all

Author List

Leading Author (Editor)			
Surname	Initials	Beneficiary Name	Contact email
Krzikalla	RK	SICK	roland.krzikalla@sick.de
Co-authors (in alphabetic order)			
Surname	Initials	Beneficiary Name	Contact email
Camps	DC	i2CAT	daniel.camps@i2cat.net
Carmona	EC	I2CAT	estela.carmona@i2cat.net
Darroudi	MD	Neutroon	mahdi.darroudi@neutroon.com
Guerra	RG	Neutroon	rolando.guerra@neutroon.com
Palomares	JP	I2CAT	javier.palomares@i2cat.net

Reviewers List

List of reviewers (in alphabetic order)			
Surname	Initials	Beneficiary Name	Contact email
López	OL	S21Sec	olopez@s21sec.com
Borne	RB	S21Sec	rborne@s21sec.com
Ubis	FU	Visual Components	fernando.ubis@visualcomponents.com
Khodashenas	PK	HWE	pouria.khodashenas@huawei.com
Krendzel	AK	HWE	andrey.krendzel@huawei.com

Document History

Document History			
Version	Date	Author	Remarks
0.1	01.04.2023	R. Krzikalla	Table of Content
0.2	01.09.2023	R. Krzikalla	First version of the deliverable
0.3	20.09.2023	D. Camps, E. Carmona, J. Palomares, M. Darroudi, R. Guerra	Draft of chapter Integration finished
0.4	06.10.2023	R. Krzikalla	Draft of hard- and software description finished
0.5	13.10.2023	O. López, R. Borne, F.Ubis	Internal review of first draft
0.6	21.10.2023	R. Krzikalla	Implement feedback from internal review
0.7	24.10.2023	P. Khodashenas, A. Krendzel	Internal review
0.8	25.10.2023	R. Krzikalla	Implement feedback from internal review
1.0	01.11.2023	R. Krzikalla	Final submission

DISCLAIMER OF WARRANTIES

This document has been prepared by Zero-SWARM project partners as an account of work carried out within the framework of the contract no 101057083.

Neither Project Coordinator, nor any signatory party of Zero-SWARM Project Consortium Agreement, nor any person acting on behalf of any of them:

- makes any warranty or representation whatsoever, express, or implied,
    - with respect to the use of any information, apparatus, method, process, or similar item disclosed in this document, including merchantability and fitness for a particular purpose, or
    - that such use does not infringe on or interfere with privately owned rights, including any party's intellectual property, or
- that this document is suitable to any particular user's circumstance; or
- assumes responsibility for any damages or other liability whatsoever (including any consequential damages, even if Project Coordinator or any representative of a signatory party of the Zero-SWARM Project Consortium Agreement, has been advised of the possibility of such damages) resulting from your selection or use of this document or any information, apparatus, method, process, or similar item disclosed in this document.

Zero-SWARM has received funding from the European Union's Horizon Europe research and innovation program under grant agreement No 101057083. The content of this deliverable does not reflect the official opinion of the European Union. Responsibility for the information and views expressed in the deliverable lies entirely with the author(s).

Executive Summary

In many industrial applications, process data must be recorded directly at the point of action, in some cases pre-processed and transmitted to a central unit. A major part of these tasks is performed by intelligent sensors with built-in computing units and data interfaces. However, more complex applications, such as in intralogistics or industrial production, require networked sensor systems that can record and process data over a wider range. For those applications the computing power of the individual sensors might sometimes insufficient to perform the desired pre-processing directly at the measurement location. For this purpose, edge computing devices are used, which on the one hand connect/network the sensors and on the other hand provide sufficient computing power for the required application purposes.

It is not always possible or economical to use wired sensor systems. However, the use of 5G communication can be used in industrial processes as an alternative to wired data transmission systems. The availability of private 5G networks opens new possibilities for industrial process design for the user.

However, to optimally integrate local sensor networks into 5G networks, gateways between wired sensor networks and the 5G networks are needed. At the same time, these gateways should also provide additional computing power for local data preprocessing.

This document describes the first steps of the development of a 5G-capable edge computing device that enables easy integration of local sensor networks into existing 5G networks and provides additional computing power to perform application-specific local sensor data preprocessing.

Table of Contents

List of Figures

List of Tables

List of Acronyms

Acronym	Description
3GPP	3rd Generation Partnership Project
4G	Fourth-Generation Mobile System
5G	Fifth-Generation Mobile System
API	Application Program Interface
ATSSS	Access Traffic Steering, Switching and Splitting
CAN	Controller Area Network
CNI	Container Network Interfaces
CPE	Customer Premises Equipment
EMC	Electromagnetic Compatibility
IEC	International Electrotechnical Commission
MPTCP	Multi-Path Transmission Control Protocol
OS	Operating System
OT	Operational Technology (OT Network)
RAT	Radio Access Technology
REST	Representational State Transfer
SIM	Subscriber Identity Module
SOM	System-on-Module
TCP	Transmission Control Protocol
TDC	Telematic Data Collector
TSN	Time Sensitive Network
VLAN	Virtual Local Area Network
WLAN	Wireless Local Area Network
USB	Universal Serial Bus
VM	Virtual Machine
VPN	Virtual Private Network

1 Introduction

In industrial automation processes, edge computing devices are essential bridges between the different layers of sensors, machines, processes, and tasks. Sensor data must be transformed into information needed for optimal control of the processes. The edge computing device must first provide enough interfaces for the physical connection of sensors, secondly the possibility of data exchange with other devices through communication, and thirdly, enough computing power for transforming the collected data into the desired information through suitable algorithms.

In this context, wireless connections offer far-reaching and flexible possibilities for integrating such a device into various industrial plants. In particular, the upcoming possibilities of the 5G technology provide new approaches to industrial automation solutions.

This document describes the first steps of the development of a 5G-capable edge computing device that enables easy integration of local sensor networks into existing 5G networks and provides additional computing power to perform application-specific local sensor data preprocessing.

The edge computing device presented in the document corresponds to the Access Domain, UE/modem item in figure 11 of D2.2 of the project.

1.1 Purpose of the document

This deliverable aims to describe the 5G capable edge computing device in the context of planned trials in Zero-SWARM to meet the future zero-emission requirement in industrial production processes. This document describes the requirements of the 5G capable edge computing device and the planned hard- and software architecture.

1.2 Structure of the document

The document is structured as follows:

- **Chapter 1** is an introduction to the whole document, describes its scope and purpose, as well as its structure.
- **Chapter 2** collects the requirements and the specifications of the planned 5G capable edge computing device.
- **Chapter 3** describes the hardware concept, the mayor components, the interfaces, and the planned housing of the device.
- **Chapter 4** describes the planned software, that runs on the device. It includes the general software structure, the operating system, and the mechanism to deploy additional software services.
- **Chapter 5** describes an approach to combine 5G and WiFi connections for edge computing devices.

2 Requirements

Edge computing devices are an essential part of industrial automation processes. Currently there are various devices for different application fields available. A major requirement for all devices is the type of integration into the existing infrastructure. Other general requirements include the type of communication interfaces, wired or wireless, the computing power, and the support for further software functions.

To specify an edge computing device that can be integrated into the newly developed 5G networks in Zero-SWARM, future requirements for industrial automation processes have been collected between Zero-Swarm partners. Several interviews between industrial partners, trial leaders, and current users of available edge computing devices have been the source for compiling the requirements. An essential summary of the collected hardware, software, and non-functional requirements are as follows (in alignment with D2.1 of the project where the same requirements have been presented with less details):

Hardware requirements:

- Multi-Core 64-Bit-CPU
- RAM >= 2GB
- Accessible Flash Memory (>16GB)
- Wireless interfaces
    - 4G/5G with global coverage
    - WLAN
    - Bluetooth
- Wired interfaces
    - Gigabit LAN up to 10GBit
    - Serial (RS232 / RS485)
    - Digital IOs with power monitoring capabilities
    - CAN
    - USB
    - IO-Link
- GPS
- Acceleration sensors
- (Micro-) SD card (external accessibility)
- SIM card (external accessibility) / eSIM
- Fanless housing
- Industrial connectors (M12)
- Industrial voltage range (9-30V)

Software requirements:

- Updateable operating system according to state-of-the-art cybersecurity policies
- Web-based configuration interface for
    - Network (Ethernet, WLAN)
    - 4G/5G (APN)

    - Digital IO
    - CAN
    - IO Link
    - Serial Interfaces
- Docker/K3S orchestration for deployment for cloud-native services
- Integrated software services (optional)
    - VPN service
    - MQTT broker and client
    - Time synchronization service (NTP, PTP, GPS)
    - Node-RED
    - OPC-UA
- WLAN access point capabilities
- Network routing capabilities
- Accessible for custom services
- API for scripted data access

Non-functional requirements:

- Compliant to typical industrial EMC- and environmental-standards
- Compliant to cyber security requirements (IEC 62443 SL2)
- Easy to integrate in existing environments and networks

3 Hardware description

Based on the described requirements and continuously collected customer feedback from the current 4G-capable edge computing device TDC-E [1] a new 5G-capable edge computing device is designed.

The following chapter describes the planned hardware concept of the new device.

The general concept of the new 5G-capable edge computing device is depicted in Figure 1. The core unit is an ARM-based System-on-Module (SOM) with onboard RAM and Flash-Memory. The wireless connections will be realized via separate modules for 5G and for WiFi/Bluetooth. Gigabit interfaces, digital IOs, serial connections (RS232/RS482), CAN-Bus, and USB are foreseen for wired connections. A SIM card slot will be available to include the needed SIM cards to connect to the 5G networks. The possibility of eSIMs is currently in discussion. Besides the described interfaces, further interfaces, such as IO-Link, are also planned.

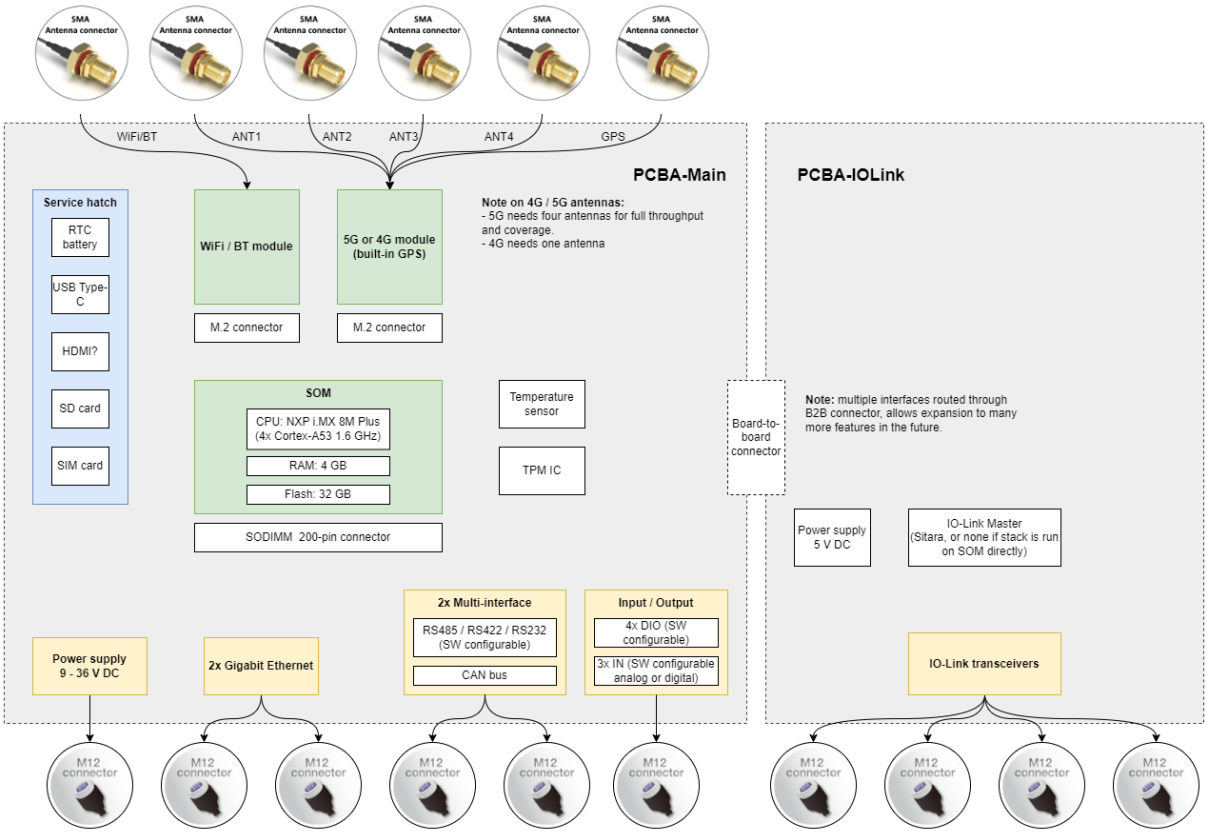

Figure 1: General concept of the new 5G-capable edge computing device

Figure 2 shows the housing design of the future 5G capable edge computing device TDC (Telematic Data Collector). The design is based on the predecessor model but is supposed to offer industrial requirements like IP67 and easy access to SIM and SD cards. A first hardware realization is planned for November 2023.

Figure 2: Housing design of the future 5G capable edge computing device TDC

4 Software description

The software architecture of the edge computing device is depicted in Figure 3. This architecture aligns with the described base model for industrial automation (ISO/IEC 30164) in work package 2, especially with Zero-SWARM deliverable D2.2 [2] (see Figure 4).

The comms section of the model refers to the capability of supporting the data exchange required by the industrial automation functionalities supported by the entity (e.g., network stacks). This is realized in the TDC hardware and the OS. The storage section refers to the capability of storing and aggregating configuration and process data (including events) related to the industrial automation functionalities of the compute section. The operating system will support both sections. Finally, the compute section refers to the capability of executing logical actions to support one or more industrial automation functionalities. This is realized by various services that are based on Docker.

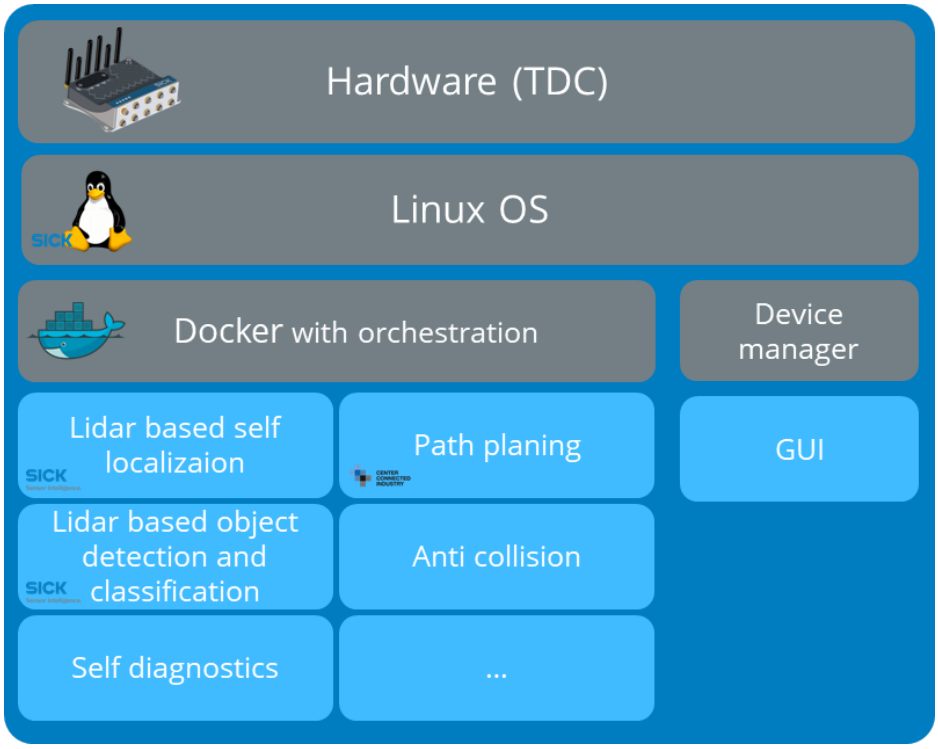

Figure 3: General structure of the micro services on the 5G capable edge computing device "TDC"

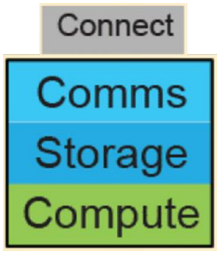

Figure 4: Base model for industrial automation (ISO/IEC 30164) [2]

In the following the mentioned sections will be described in more detail.

4.1 Operating system

The operating system is Linux-based and self-implemented for the used hardware. The structure is shown in Figure 5. The structure is explained in section 4.2 in detail. The Kernel version is 4.19 with specific patches for a full hardware support. An update to Kernel version 5.10 is planned for the second half of 2024.

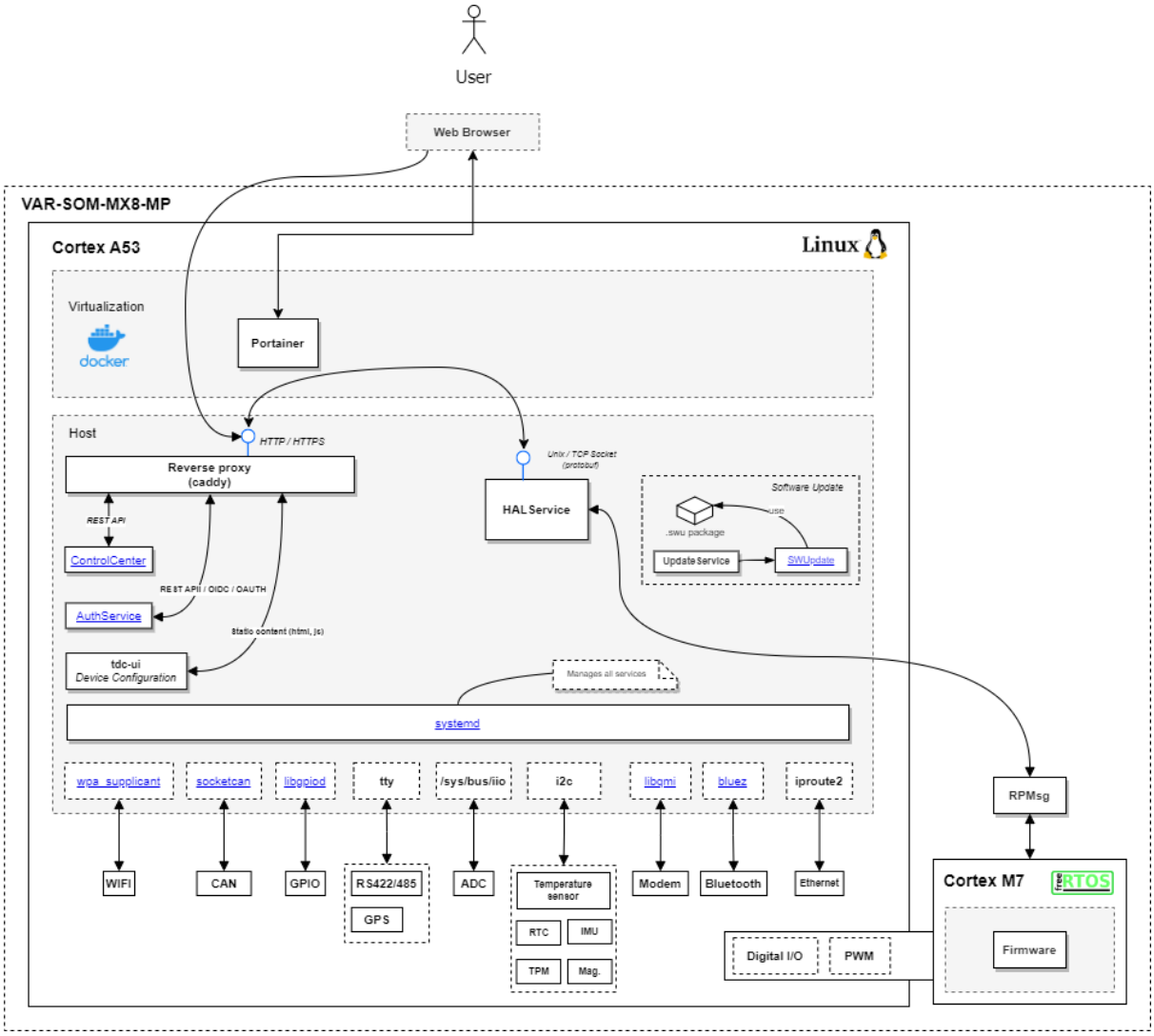

Figure 5: Structure of the operating system of the TDC

4.2 Structure of services

In the user space several standard components are included that will be described shortly in Table 1.

Table 1: Collection of used components and services in the user space

Service	Description
systemd	*systemd* is a suite of basic building blocks for a Linux system. It provides a system and service manager and starts the rest of the system.
libqmi	*libqmi* is a glib-based library for talking to WWAN modems and

	devices which speak the Qualcomm MSM Interface (QMI) protocol.
BlueZ	*BlueZ* is a Bluetooth protocol stack for Linux.
libgpiod	*libgpiod* - This library encapsulates the ioctl calls and data structures behind a straightforward API.
socketcan	The *socketcan* package is an implementation of CAN protocols (Controller Area Network) for Linux.
ASP.NET Core Runtime	*ASP.NET Core* and *.NET* runtime is optimized for running ASP.NET Core apps in production.
SWUpdate	*SWUpdate* is a Linux Update agent with the goal to provide an efficient and safe way to update an embedded Linux system in the field. *SWUpdate* supports local and OTA updates, multiple update strategies and it is designed with security in mind.
wpa_supplicant	wpa_*supplicant* is a cross-platform supplicant with support for WPA, WPA2 and WPA3 (IEEE 802.11i). It is suitable for desktops, laptops and embedded systems. It is the IEEE 802.1X/WPA component that is used in the client stations. It implements key negotiation with a WPA authenticator, and it controls the roaming and IEEE 802.11 authentication/association of the wireless driver.
Docker	*Docker* is a set of platform-as-a-service products that use OS-level virtualization to deliver software in packages called containers.
Portainer	*Portainer Community Edition* is a lightweight service delivery platform for containerized applications that can be used to manage Docker, Swarm, Kubernetes and ACI environments.

The depicted services on top of the Docker service (Lidar-based self-localization and mapping, Environment perception, path planning, Anti-collision, etc.) are part of work package 6.2 and will be described in detail in deliverable D6.1 [3] and D6.2 [4].

4.3 Custom network interfaces

The use of Docker containers has already been demonstrated in industrial applications. Containerization technologies can facilitate emerging use cases by facilitating the deployment of micro-services across multiple hosts. However, industrial applications have specific networking requirements, often requiring layer 2 communication. Kubernetes [5] has become the leading container orchestrator due to its open-source nature, vendor neutrality, scalability, extensive ecosystem, and community support, making it a versatile and widely adopted solution for containerized application management. However, Kubernetes deploys an IP network mesh among pods[1], which is usually inadequate for industrial applications [6]. However, new networking plug-ins have become available in the Kubernetes ecosystem, which enables the configuration of additional

[1] Kubernetes pods are groups of one or more containers, with shared storage and network resources, and a specification for how to run the containers.

network interfaces among micro-services (i.e., pods), such as bridge, macvlan, and so on. This section provides an in-depth exploration of interface definition using the tool Multus CNI 21 [7] and its underlying logic within the realm of container-based architectures for industrial micro-services. Its central focus is on provisioning the essential framework for configuring and overseeing the networking module. The objective is to enable multiple VLAN-based communications between Pods inside the K8s cluster. This capability provides heightened versatility in network configuration and allows for effective traffic isolation.

Multus CNI is a Kubernetes plug-in for container network interfaces (CNI), facilitating the attachment of multiple network interfaces to pods. Unlike the usual setup in Kubernetes where each pod is equipped with a single network interface (in addition to a loopback), Multus empowers the creation of multi-homed pods that can possess several interfaces. This functionality is achieved through Multus functioning as a "meta-plugin," which is a CNI plug-in capable of invoking various other CNI plug-ins.

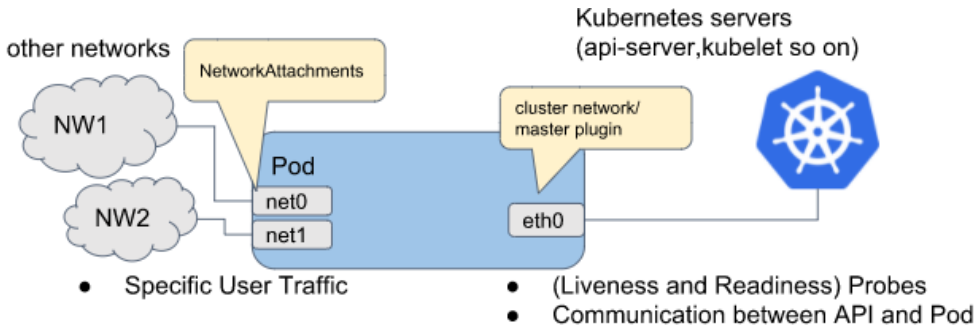

Figure 6: Diagram showing network interfaces attached to a pod, provisioned by Multus CNI [7]

Figure 6 illustrates the process by which Multus CNI provisions additional network interfaces that are then attached to a pod. It possesses three interfaces: eth0, net0, and net1. The interface eth0 establishes a connection between the Kubernetes cluster network and Kubernetes server/services (such as the Kubernetes API server, kubelet, etc.). Moreover, the interfaces net0 and net1 serve as supplementary network attachments, connecting to other networks through the utilization of different CNI plug-ins (for example, vlan/vxlan/ptp).

In the proposed scenario, an industrial application comprised of two micro-services is deployed over one customer-provided equipment (CPE) and one virtual machine (VM) running in an industrial edge, each responsible for allocating a micro-service (pod). These pods are interconnected using two distinct interfaces, determined by the technology employed to transmit the information, be it 5G, Wifi, or Ethernet. Figure 7 depicts the architecture of the proposed setup, where policies are set in both devices to control the data flow between them.

In a future implementation stage, this concept will be showcased through the implementation of a multi-RAT use case, where wireless transmitted data can be routed through WiFi, 5G or aggregated it using the multi-path TCP (MPTCP); in such case scenario, the required connectivity service interfaces for each radio technology will be implemented with Multus CNI. Moreover, this case scenario is complementary to the multi-RAT concept explained below in section 5.1.

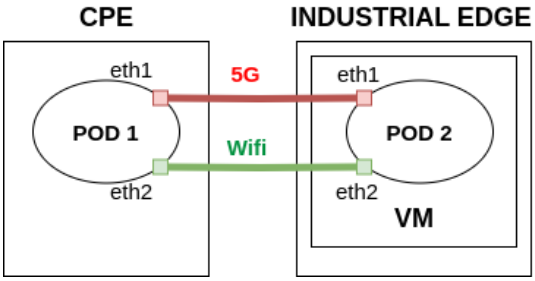

Figure 7: Architectural diagram showing the interconnection through two distinct interfaces.

5 Integration

This Section describes an approach to combine 5G and WiFi connections for edge computing devices.

5.1 5GNR and Wi-Fi6 access networks integration for 5G edge computing device

A solution to integrate 5GNR and WiFi6 access networks under a common system has been proposed by the 5G-CLARITY project [9]. 5G-CLARITY proposes a multi-connectivity framework that is based on the Adaptive Traffic Steering, Splitting, and Switching (ATSSS) defined by 3GPP in R16.

The blue components in Figure 8 depicts the 5G-CLARITY solution consisting of:

- A CPE that includes both 5GNR and WiFi radio interfaces.
- A private 5G infrastructure consisting of (a set of) 5GNR gNB(s) and WiFi AP(s). gNBs connect to the core network, which provides access to a Data Network Name (DNN), which can also be accessed through the WiFi APs. Note that the WiFi interface in the CPE acquires an IP address from a DHCP (Dynamic Host Configuration Protocol) server deployed in the DNN, whereas the 5GNR interface acquires an IP address from the core network.
- In the CPE and DNN, a user plane function is deployed that can aggregate data traffic from the WiFi or the 5GNR access networks. This user plane function is based on Multi-Path TCP (MPTCP) [10], which spans two dedicated TCP sub-flows, one through the 5GNR access and one through the WiFi access network, between the CPE and an MPTCP proxy function deployed in the DNN. The MPTCP user plane proxy allows to schedule the data transmitted through each TCP sub-flow according to three available scheduling modes:
    - Default scheduling: Whereby the MPTCP scheduler keeps track of the RTT across each TCP sub-flow, and schedules data always across the path with the lowest delay. This scheduler compromises capacity and latency.
    - Round-robin scheduling: Whereby the MPTCP schedulers transmits segments across each flow in a round-robin fashion. This scheduler is suitable for capacity hungry services.
    - Redundant scheduler: Whereby the MPTCP scheduler duplicates each segment across all paths in parallel, while delivering the first arriving segment to the receive socket and discarding the rest. This scheduler is suitable for latency services.

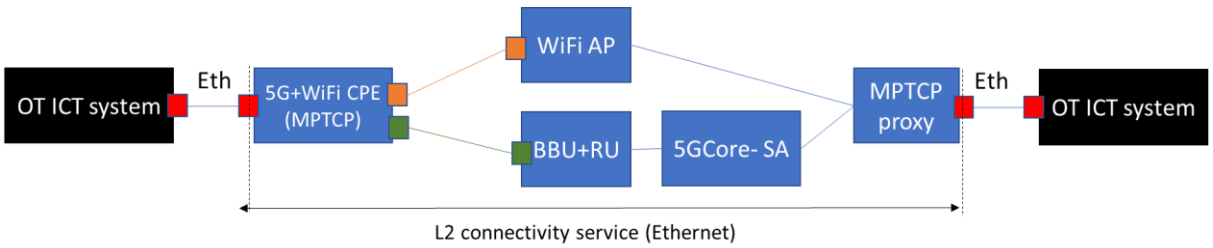

Figure 8: 5G-CLARITY concept of 5G/WiFi enabled CPE

Some constraints of the solution include:

- The MPTCP functionality is not, as of this writing, yet fully available upstream, which means that the CPE needs to support a custom Linux kernel to support MPTCP. This customized kernel may conflict with other features that are required in the CPE.
- The MPTCP proxy is an additional function that needs to be deployed behind the core network, thus increasing the complexity of the network.
- As depicted in Figure 8, in an industrial scenario, the CPE must offer the aggregated 5GNR/WiFi data pipeline to tunnel external traffic generated in the OT network, which will often be Ethernet traffic. No means are provided in the 5G-CLARITY project to tunnel Ethernet traffic over an aggregated 5GNR/WiFi solution.

In Zero-SWARM we will study means to address the previous limitations in the following way:

- The use of MPTCPv1.0 [11] will be explored. MPTCPv1.0 is a derivative from the original MPTCP implementation that is included in the Linux Kernel upstream. As of this writing, this implementation still suffers from some limitations with respect to the original MPTCP implementation. Progress of this implementation will be monitored during Zero-SWARM to see if it becomes a viable option for the Zero-SWARM 5G/WiFi CPE.
- To minimize the impact of deploying the MPTCP proxy as an additional network function, a virtualized version of the MPTCP proxy will be onboarded on the edge server included in the NOMAD private 5G solution developed by NEUTROON in T3.2.
- To enable the transport of Ethernet services over the aggregated 5G/WiFi access networks, the work on end-to-end L2 service provisioning in T3.2 will be extended to operate on top of MPTCP.

6 Conclusion & further work

In this document the concept and the current state of the development of a 5G-capable edge computing device have been described. The new aspect here is the industrial need of the combination of a 5G gateway, that provides the seamless connection of local wired sensor networks with the wireless 5G networks, and an edge computing device, that brings the possibility to pre-calculate local data for further application development. In the further course of the project, prototypes of the described 5G capable edge computing device will be built-up and tested. This is planned for the first and second quarter of the last project year. The tests include general hardware tests (EMC, vibration, housing, etc.), interface tests and tests of the included software (operating system, service programs, application programs, etc.) according to the described tests in D6.1 [3] and D6.2 [4]. At the end of this work package, a 5G-capable edge computing device should be available that can be used for industrial applications, e.g. in the fields of intralogistics or industrial production. In the context of ZeroSWARM, it will also be actively used in two different trials (Trial 1 and Trial 3 in the Central Node).

Furthermore, we will evaluate how the mechanisms to transport L2 services designed in Task 3.2, perform when transported over a CPE that is capable of aggregating 5G+WiFi.

References

[1] Telematic data collector TDC-E https://www.sick.com/de/en/c/g444354?tab=overview, accessed on 06.10.2023.

[2] Zero-SWARM Deliverable D2.2 - Eco designed architecture, specifications & benchmarking.

[3] Zero-SWARM Deliverable D6.1 - Integration, validation, specification of the trial demonstrations

[4] Zero-SWARM Deliverable D6.2 - Trial demonstration and evaluation results

[5] https://kubernetes.io/de, accessed on 25.10.2023.

[6] Gundall, M., Reti, D. and Schotten, H. D., "Application of Virtualization Technologies in Novel Industrial Automation: Catalyst or Show-Stopper?" 2020 IEEE 18th International Conference on Industrial Informatics (INDIN), Warwick, United Kingdom, 2020, pp. 283-290.

[7] "Multus-CNI Documentation", https://github.com/k8snetworkplumbingwg/multus-cni, accessed on 19.09.2023

[8] "Nmstate: A Declarative Network API", https://github.com/nmstate/nmstate, accessed on 19.09.2023

[9] Cogalan, T. et al., "5G-CLARITY: 5G-Advanced Private Networks Integrating 5GNR, WiFi, and LiFi," in IEEE Communications Magazine, vol. 60, no. 2, pp. 73-79, February 2022, doi: 10.1109/MCOM.001.2100615.

[10] Ford, A., Raiciu, C., Handley, M., Bonaventure, O., & Paasch, C. (2020). RFC 8684: TCP Extensions for Multipath Operation with Multiple Addresses.

[11] https://www.multipath-tcp.org/, accessed on 20.09.2023.