D4.3

REPORT ON THE DEVELOPMENT OF 5G USE CASES

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

D4.3 Report on the development of 5G use cases

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:	R: Document, report
Deliverable reference number:	D15
Contributing workpackages:	WP4
Dissemination level:	Public
Due date:	May 31, 2021
Actual submission date:	May 31, 2021
Responsible organization:	ERI-HU
Editor(s):	Norbert Reider (ERI-HU)
Version number:	V1.0
Status:	Final

Short abstract:	This deliverable contains a detailed description of the developed and finalized use cases including final end-to-end integrated systems for Cloud-based mobile robotics, as well as, for TSN/Industrial LAN over 5G system.

Keywords:	Mobile robots, AGVs, 5G, cloud control, collaboration, Industrial LAN over 5G

Contributor(s):	Norbert Reider (editor, ERI-HU) Gabor Nemeth (ERI-HU) Sandor Racz (ERI-HU) Attila Vidacs (BME) Gabor Feher (BME) Markosz Maliosz (BME) Harutyunyan Davit (Bosch) Peter Buseck (Bosch) Leefke Grosjean (ERI-SE) Joachim Sachs (ERI-SE)

Disclaimer

This work has been performed in the framework of the H2020 project 5G-SMART co-funded by the EU. This information reflects the consortium's view, but the consortium is not liable for any use that may be made of any of the information contained therein.

This deliverable has been submitted to the EU commission, but it has not been reviewed and it has not been accepted by the EU commission yet.

Executive summary

This deliverable describes the use case architectures and the status of the implementation of the 5G-SMART use cases described in 5G-SMART Deliverable 1.1 [5GS20-D110], namely *Cloud-based mobile robotics (collaborative AGVs)* and *TSN/Industrial LAN over 5G network*. They will be deployed in the semiconductor plant of Bosch in Reutlingen, Germany. We discuss the developed plan for the collaborative AGVs use case, where we use a commercially available AGV and a custom designed one (called research AGV) to realize the collaboration. In the report we explain how the intelligence of the research AGV is removed, reimplemented and extended in a cloud native manner to be executed from an edge cloud environment. The commercial AGV is also discussed which is connected over 5G, and part of the control logic is relocated to factory cloud. We talk about how the hardware elements including the mobile base platform of the research AGV, necessary sensors, as well as, the commercial AGV are selected and implemented. We present the hardware and software architectures designed, as well as, the integration of the sensors, their drivers and the 5G modems into the AGVs' architecture. The communication diagram of the detailed software architecture is also included and explained in this report. The cloudification part of the work is discussed where we explain the reliability and scaling cloud features.

Regarding the second use case, i.e., the TSN/Industrial LAN over 5G, we present the completed development of the end-to-end system architecture. Additionally, we describe the chosen hardware components to be used for implementing and validating this use case. This is followed by presenting the status of the integration and the end-to-end testing of the hardware components for the industrial controller-to-controller communication, as well as, for Rudolph F30 industrial machine to its backend server communication over the 5G network.

Contents

Disclaimer ... 1
Executive summary ... 2
1 Introduction ... 4
1.1 Cloud-based mobile robotics in factories ... 4
1.2 TSN/Industrial LAN over 5G in the shop-floor ... 4
1.3 Structure of the document ... 5
2 Cloud-based mobile robotics in factories ... 6
2.1 Preparations ... 6
2.2 System design ... 6
2.2.1 Research AGV ... 7
2.2.2 Commercial AGV ... 9
2.2.3 Software and hardware architectures ... 10
2.2.4 Factory cloud aspects ... 17
2.2.4.1 Benefits of the factory cloud ... 17
2.2.4.2 Factory cloud deployment ... 17
2.3 End-to-end system integration ... 19
3 TSN/Industrial LAN over 5G in the shop-floor ... 20
3.1 End-to-end system architecture and use case ... 20
3.1.1 Hardware and software components ... 21
3.2 End-to-end system integration ... 24
4 Summary and future work ... 25
Appendix A: Selection of AGV platforms ... 26
Appendix B: Certifications ... 28
Appendix C: Sensors for the research AGV ... 28
List of abbreviations ... 30
References ... 31

1 Introduction

The objective of the document is to provide a detailed description of the 5G use cases that are going to be trialed at 5G-SMART's trial site in Reutlingen. We explain the architecture and the progress of the implementation of the two 5G use cases to be validated in the semiconductor factory of Bosch. The first use case is about cloud-based mobile robotics, which focuses on the flexible transportation in modern factories. The second use case investigates how 5G technology can be used to transport the traffic of Time Sensitive Networking (TSN)/industrial LAN networks. A detailed description of the use cases, requirements and Key Performance Indicators (KPIs) can be found in 5G-SMART's Deliverable D1.1 [5GS20-D110], below only a short summary is given and further details are provided in separate sections (Sections 2 and 3). The description of the trial site, use cases and trial site constraints, as well, as the wireless 5G infrastructure can be found in 5G-SMART's Deliverable D4.1 [5GS20-D410]. A short introductory video to the use cases can be found on 5G-SMART's YouTube channel [5GS21-YT].

1.1 Cloud-based mobile robotics in factories

This use case focuses on the feasibility, flexibility, and performance of wirelessly controlled Automated Guided Vehicles (AGVs) in a manufacturing shop-floor equipped with 5G technology. We note that the AGVs used in this work package do not rely on any guidance such as colored or magnetic stripes on the floor or any other marker, they are fully autonomous mobile robots. Besides the need for low-latency and reliable radio connectivity provided by 5G as an enabling technology, one novelty of this use case is the possibility to decouple the closed-loop control of the robot from the robot's embedded system and place it into an edge cloud execution environment (i.e., a factory cloud, please see 5G-SMART's terminology document in [5GS20-CT]) while sustaining the KPIs, like sufficiently low execution latency and adequate fault-tolerance. Moving the control logic into the cloud benefits from scaling of the workload when changing the tasks for the robots, ease of maintenance of the control software and improved resiliency to software and hardware failures. Furthermore, decoupling the control logic from the AGV enables innovative control solutions such as collaboration between individual AGVs by, e.g., facilitating the creation and sharing of up-to-date common maps. For instance, simultaneous localization and mapping (SLAM) capabilities of an AGV can enhance the route selection for other AGVs in real-time, i.e., one AGV detects an obstacle, the other one reacts by finding another path to the destination.

1.2 TSN/Industrial LAN over 5G in the shop-floor

This use case focuses on investigating and validating the applicability of 5G for transporting the traffic of TSN/industrial LAN (I-LAN) applications. Nowadays, due to the stringent requirements of the industrial applications, all operational I-LANs are realized based on fixed (wired) communication networks. Limited flexibility for setting up new production lines or for restructuring an existing production line, as well as complex and costly maintenance, are major drawbacks of the wired I-LAN realizations. In particular, this can be an issue in view of the recent trends for making the industrial environments as flexible as possible, e.g. smart factories of the future in the context of Industry 4.0.

This use case mostly focuses on evaluating the feasibility of partially replacing fixed interconnections between TSN/I-LAN nodes with 5G mobile communications, which, due to its low-latency characteristics, is considered to be a good choice for satisfying the stringent requirements of the industrial applications in terms of latency and reliability. This will reduce the cables and connectors wear and tear, for the mobile machines/controllers, resulting in reduced maintenance costs. Additionally, replacing the cables for communications between controllers and machines with 5G communications results in a greater flexibility for implementation and adaptation of the industrial manufacturing infrastructure. Consequently, this can improve the productivity of manufacturing through reducing the time for setting up or customizing a production cell/line and improving the maintenance.

1.3 Structure of the document

The document is structured as follows. After the introduction, Section 2 and 3 present the planning, the system design, as well as, the implementation details including the end-to-end system integration of the cloud based mobile robotics and the TSN/Industrial LAN over 5G use cases, respectively. The report ends with a summary and outlook of future work (Section 4).

2 Cloud-based mobile robotics in factories

2.1 Preparations

In this subsection, we describe some preparation details that are important from system design and implementation perspectives. Based on the use case analysis in terms of requirements and KPIs performed and described in Deliverable D1.1 [5GS20-D110], suitable hardware platforms were chosen. The hardware platforms have to support the use cases while fulfilling the functional requirements and challenges coming from the real production environment of the Bosch factory such as safety and clean room certifications.

Fulfilling all requirements at the same time turned out to be a major challenge, since typically the features of commercial AGV platforms with necessary certifications do not enable the realization of remote closed-loop control. Every component is onboard and closed, i.e., there is no available external interface to use for the cloud-based low-level control. On the other hand, there are research AGV platforms that easily support the remote execution of even the millisecond-scale control, however, mainly for the same reason (availability of low-level control interfaces), they do not have safety certifications that are needed to operate in a real production environment.

Hybrid solution

As the final approach, we decided to design a **hybrid solution** where the collaborative AGV use case is realized by **using a research AGV, as well as, a commercial AGV platform**. It means that a suitable research platform is used to show how we can leverage 5G and cloud technologies to enable novel collaborative control solutions based on the cloud-native realization of the AGV control. The certified commercial AGV platform is connected over the 5G system to show the benefit of collaborative knowledge collected in the factory cloud (e.g., using the common map for trajectory planning). Since the commercial platforms are typically closed, changing the connection technology to 5G, as well as, applying customized high-level control instead of their legacy (closed) software packages are already a challenge. On the other hand, commercial platforms are better suited for demonstration and testing in real factory environment and can be deployed on the shop-floor.

As the final choice, we selected **MiR 100 as the commercial platform** and a **custom AGV based on the HEBI Mobile Base [HEBI] as the research AGV**. We have a detailed description on the selection of robot platforms in the Appendix A. These AGVs are either completely CE certified (MiR 100) or built up using CE compliant components (research AGV). However, building from CE certified components does not mean that the whole platform will be CE certified, so we investigated the possibilities to certify the research AGV and the results are included in Appendix B.

2.2 System design

In this subsection we describe the details on the hardware and software design we realized in the collaborative AGV use case. As discussed above, we concluded to use a research and a commercial AGV platform to implement the collaborative AGV control. With this in mind, we introduce some details of the hardware design and software development, as well as, illustrate the final hardware and software architectures.

The hardware components and features are discussed in separate subsections regarding the research and commercial AGVs. The integration of the two heterogeneous systems is described in the software architecture part.

2.2.1 Research AGV

We selected the HEBI Mobile Base as the research platform which we extended to realize an AGV (or autonomous mobile robot as often called nowadays). In Figure 1, we illustrate one of their robotic kits that we used as a starting point. We tailor-made this mobile base for the needs of our use case. This mobile base kit consists of four wheels, four servo motors, and a chassis that holds a box including the batteries and some electronics such as an Intel NUC mini PC, and voltage converters.

It is important to note that this platform is fully customizable in every possible way including the mechanical design and the hardware components, which is exploited to create a platform that is best suited for the given factory environment. The final hardware was designed iteratively, and several CAD drawings were made, of which one example is shown in Figure 2.

There are two types of servo motors as also shown in Figure 1, X-Series and R-series servos. Each servo module is a series-elastic actuator that integrates a brushless DC motor, gear reduction, force-sensing, encoders, and control electronics into a compact package and communicates using standard 10/100Mbps Ethernet. The actuator is designed to function as a full-featured robotic component as opposed to a simple servo motor, for instance, the actuator can rotate its output continuously and it requires no calibration or homing on boot-up. The modules can be used in everything from wheeled robots to multi-degree-of-freedom collaborative robotic arms. The APIs of the HEBI servos provide different levels of access to the hardware. Kinematics and trajectory APIs provide high-level control, similar to that of today's industrial robots. Joint-level control APIs allow control of each actuator's motion and tight integration of feedback from multiple actuators and external sensors.

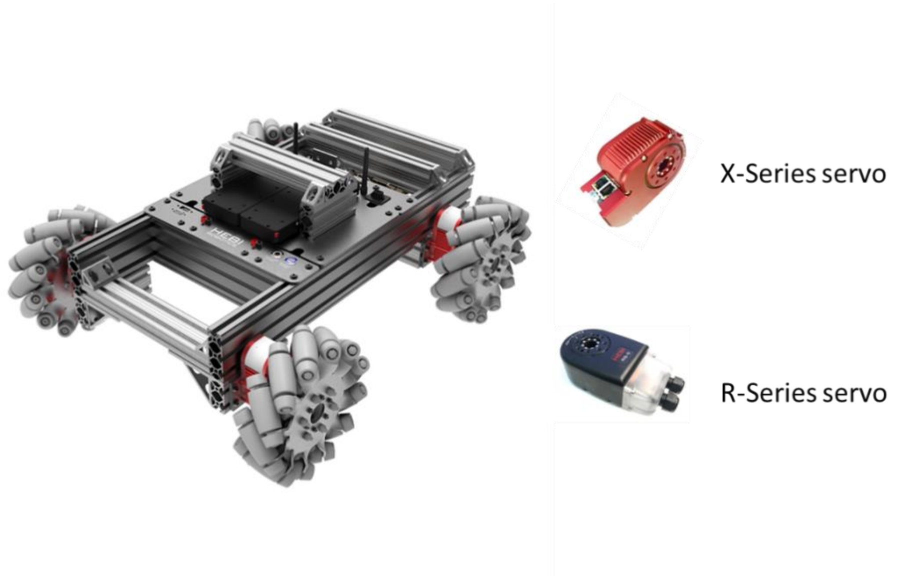


Figure 1 HEBI Mecanum Drive Mobile Base Kit

The R-Series actuators are the sealed versions (IP67) of the X-Series and designed with a lightweight form factor that allows them to be used in challenging field applications. We selected the R-Series actuators to be used as wheel drives for the platform to be compliant with cleanroom requirements. Since each actuator has a brushless DC motor that is boxed into an IP67 frame, dust is not expected to be generated at all.

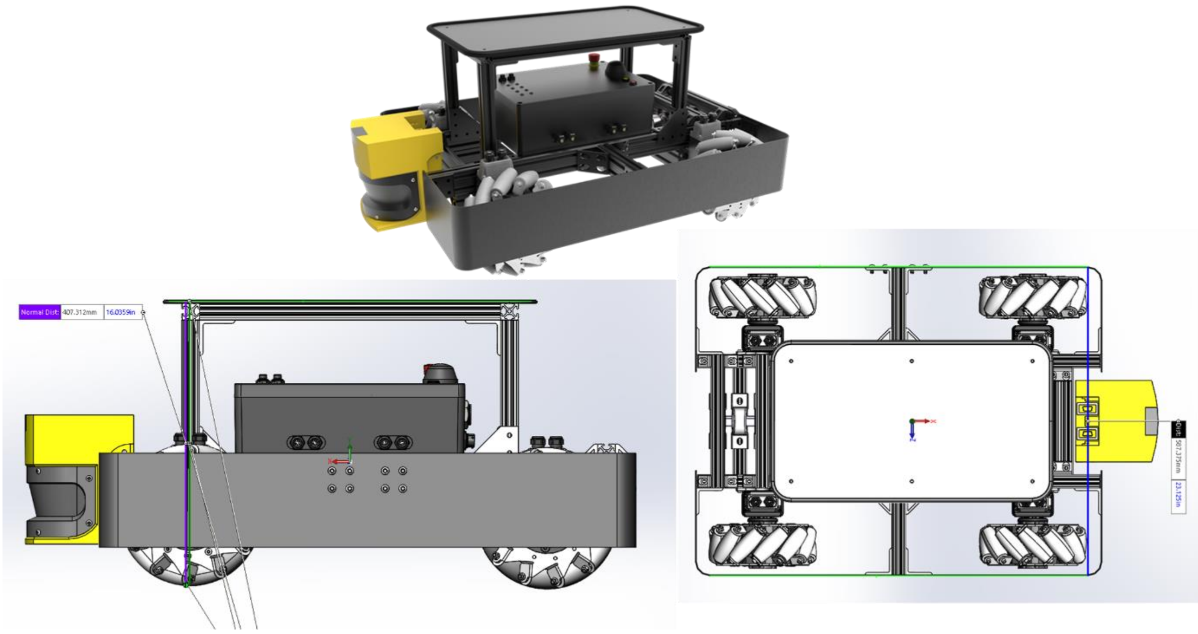


Figure 2 CAD drawings of the mobile platform for the research AGV, showing snapshots of the design process

As illustrated in Figure 2 the system is designed to have a shield around the wheels and the whole body to help preventing the platform to get stuck during maneuvering (e.g., items cannot stuck between the wheels and the chassis). The chassis is built from Bosch-profiles to allow flexible extension of the platform, e.g., to attach sensors. A flat plate is also added on the top of the mobile base for placing the 5G modem and transportation purposes. The yellow box in the front of the platform is just a possible location of a sensor, the exact type and location of the sensors are described later.

The HEBI Mobile Base is a simple hardware with Mecanum wheels, servo motors, a chassis, a device for computation and gatewaying, some electronics for power conversions, different connectors and an emergency button as illustrated in Figure 3 with the marking of the CE certified components. Servo motors are CE certified and the remaining parts have either no electronics or are off-the-shelf components with CE marks. As the figure shows, there are no software components associated with the HEBI platform, since it is not an AGV by itself, it cannot execute autonomous movements, thus different sensors need to be attached to explore environment, provide positioning information, as well as, for integrating safety mechanisms. Furthermore, all necessary software modules need to be developed to process the sensor information and to steer the whole platform.

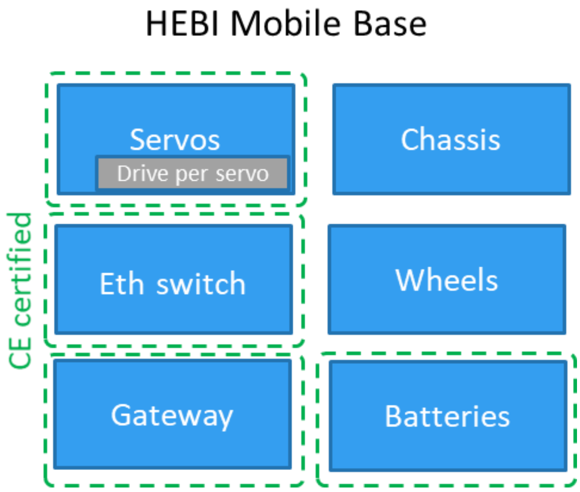


Figure 3 Abstraction of components inside the HEBI platform

Final set of sensors

For the final set of sensors to be added to HEBI Mobile Base we ordered two Sick S300 Standard S30B-3011BA safety laser scanners, which, besides their safety features such as the built-in safety zone handling, they provide the laser scan data on RS-422 serial interface so there is no need for separate lidars. We identified the need for a camera input to add 3D sensing of the environment, since the laser scanner provides 2D measurements only, thus we purchased an Intel RealSense D345i camera and attached to the front part of the research AGV. We also ordered ultrasonic sensors (Maxbotix I2CXL-MaxSonar-EZ4). However, it turned out that due to the narrow angle of vision, the benefit of these sensors is marginal over the information already being provided by the laser scanners and cameras.

2.2.2 Commercial AGV

Commercial AGVs do not have externally accessible interfaces, i.e., they are not designed to support control using any custom software module. The reason for that is mainly to prevent violation of the requirements corresponding to different certifications. However, some platforms may support an API that can access some sensor data and can be used to send high level commands to the device (for instance, “go to position A” or similar).

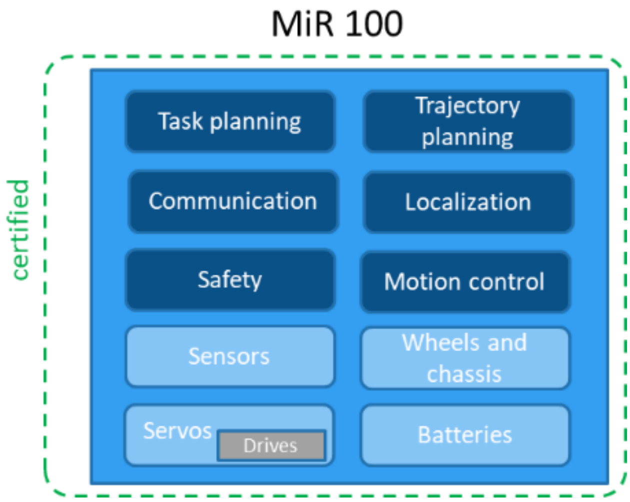


Figure 4 Abstraction of HW and SW components in the MiR 100 platform (light blue: HW, dark blue: SW components)

In Figure 4 we illustrate the simplified view of the HW and SW components of the MiR 100 AGV that we selected as the commercial platform to use. As opposed to Figure 3, the commercial platform contains all the sensors, as well as the necessary software modules onboard and the whole system is certified. However, there is no access to most of the modules from external sources.

2.2.3 Software and hardware architectures

Our solution implements the functional architecture shown in Figure 5. Most components are offloaded from the hardware device to the local factory cloud.

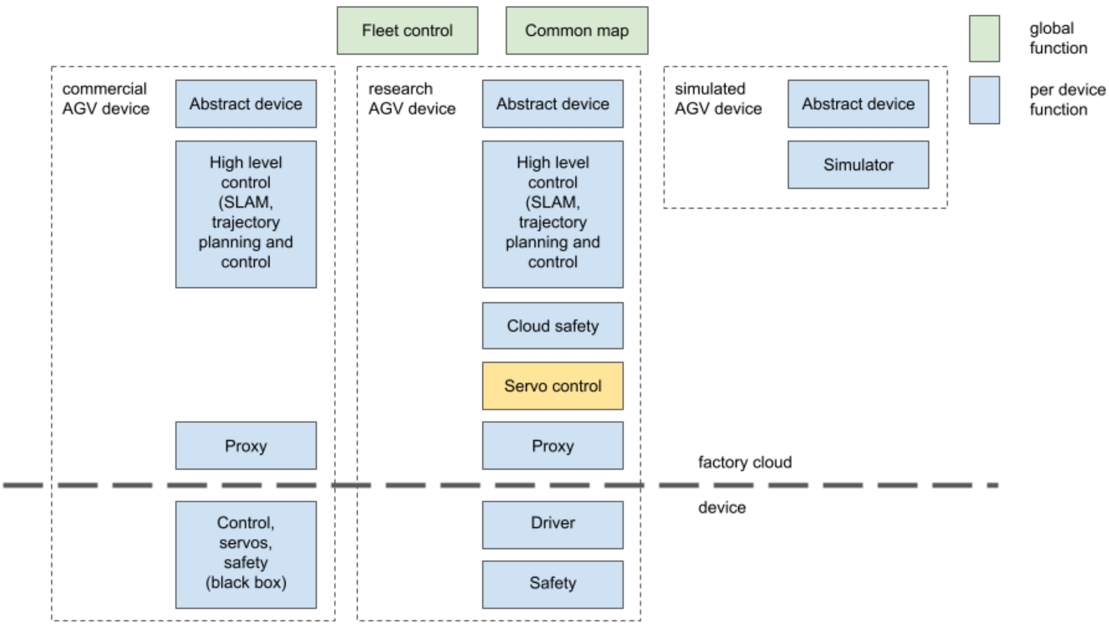


Figure 5 Functional architecture of the final realization

Figure 5 shows a stack of functional blocks for the two hardware device platforms (MiR and Research), the simulated pseudo-devices, and the global control functions at the top. Functional blocks typically communicate with their immediate upper and lower neighbors. Below the functional blocks are further explained.

Fleet control is the main business logic that users, e.g., factory plant operators interact with. It receives commands like "move a free AGV to position X within the lab". This is the highest level of the commands, and regardless of the selected devices, this level is common for both the commercial and research AGVs.

Trajectory control, as part of the **High level control** box, executes the movement of the AGV along a pre-planned path. The path is a result of a **Trajectory planning** task that uses all available information from other components to provide the most efficient physical path of movement. These functions are also referred to as *Navigation*. This information might include the actual working paths and obstructions in the workspace and the actual as well as future positions of other AGVs. Trajectory control monitors the AGV positions in real time and issues commands for the next movements.

SLAM is Simultaneous Localization and Mapping. With this component the device measures the surrounding environment and creates a map, while simultaneously it also tries to localize itself on the self-created map. When the device moves, SLAM tries to identify the new location based on the existing map and extends the map with new measurements. The source of the map is usually a point cloud, detected by LiDAR or 3D camera sensors. Usually, the resulting map is the floor plan of the area, where the device can freely move.

Common map is where the AGVs jointly store their actual views of their surroundings. The global map covers all the workspace area where the AGVs have ever been. The global map is always updated with the fresh information coming from the AGVs, so it always changes with time. Moreover, future states with the locations of the fleet can be estimated as well, when the trajectories are also known.

Safety stops the device in an emergency, i.e., when there is a risk of collision with humans, other AGVs, equipment, or walls. There could be several levels of a safety stop. In the most serious case, where the human life, the device, or surrounding items are in direct danger, the device should fully stop in order to prevent any damage. When caution is necessary, the AGV can be slowed down, as a different safety level. This way the AGV and/or others might have enough time to make a maneuver that avoids any damage. Safety is usually a separated, high reliability system on the device impacting on very low level to prevent any software or hardware errors to block the safety functions. Besides the automatic functions, safety should be available for manual triggering as well.

Cloud safety is a shadow pair of *Safety* in that it does not actually stop the device during emergencies but signals such an intent. Its output will be compared with that of S*afety*, and its purpose is to demonstrate whether safety functions could be moved to the factory cloud or not.

Servo control executes low-level commands and monitors the AGV's progress according to the path provided by *High level control*. These downlink commands towards the physical device and uplink measurements and status reports from them are relayed through a **Proxy** functionality that translates between generic and device-specific data formats. It is highlighted in the figure (in orange), since it is a real-time application which is also moved to the factory cloud for the research AGV.

Figure 6 shows the software components mapped onto the hardware architecture of the system, including the factory cloud part and the two AGV types. Typically, the controllers of the commercial AGVs are closed, except monitoring data available on standard interfaces, but without access to the internals. Still, few AGV manufacturers provide some interface, on which one can drive their device with high level commands not using their official software. In case of MiR 100, we can access sensor data and odometry information via ROS interfaces, so we can run the navigation and localization functions from the factory cloud. By odometry we mean the use of data from motion sensors to estimate change in position over time.

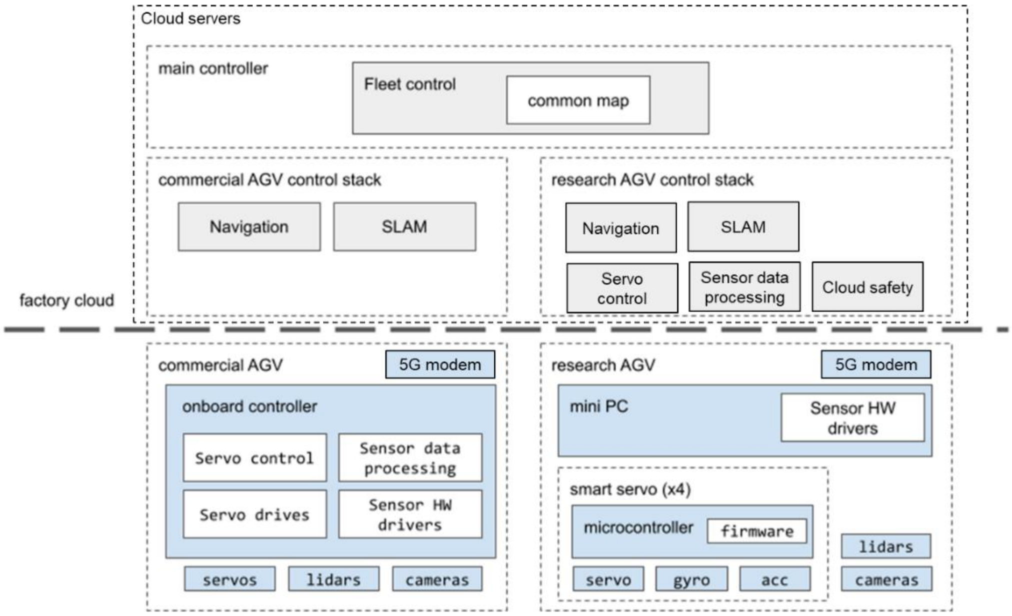


Figure 6 Software components and hardware architecture

The research AGV is built up with similar components to the commercial AGV, however as it was stated before, all the software components are open for development. The HEBI actuator has a servo motor, encoders, gyroscope and accelerometer sensors and its own controller (gyro and acc stand for gyroscope and accelerometer in Figure 6). The drivers of the sensors are running inside the actuator. The factory cloud can access the actuator and all its sensors through a communication link which goes through a mini PC residing on the research AGV. The mini PC is used only as a driver for all the sensors that are not placed inside the actuators. It is also applied as a gateway between the actuators and the 5G network. The sensors outside of the actuators are the laser scanners and the depth camera in this case.

On the top of the software architecture resides the factory cloud and the services running in it. The collaboration is realized as sharing a global (common) map built in the factory cloud, for instance, to optimize path selection and route execution.

The main controller of the research AGV also contains servo control and sensory data collection point. The sensors track the movement and the surroundings of the research AGV, while the servo control drives it. The fleet control component gathers the individual maps created by the AGVs and merges them into a global map. Knowing the global map, the actual and the planned future positions of the AGVs are the key to create the trajectory for a new AGV job or reorganize the actual routes for a more efficient solution or in the case of changes, e.g., temporary path blocks in the global map.

The actual realization of the above hardware architecture for the research and commercial AGVs are illustrated in Figure 7 and Figure 8, respectively.

Figure 7 Realized research AGV with all sensors, emergency button and a 5G industrial modem

Figure 8 MiR 100 extended with an emergency button and a 5G modem

Figure 9 shows the implemented software architecture of the system and highlights the main communication (based on TCP/IP) and control messages between the components. All the higher layer components were implemented using ROS (Robot Operating System). The Base control node is our custom implementation that realize the control of the four servos of the research AGV.

Robot navigation: The Robot navigation component implements the standard ROS 2D navigation stack that takes in information from odometry, sensor streams, and a goal pose and outputs safe velocity commands that are sent to a mobile base. The Navigation stack serves to drive a mobile base from one location to another while safely avoiding obstacles. It uses information from sensors (laser scanners and camera) to avoid obstacles in the world. It also requires that odometry information be published by the robot. The navigation stack assumes that it can send velocity commands on the "cmd_vel" topic.

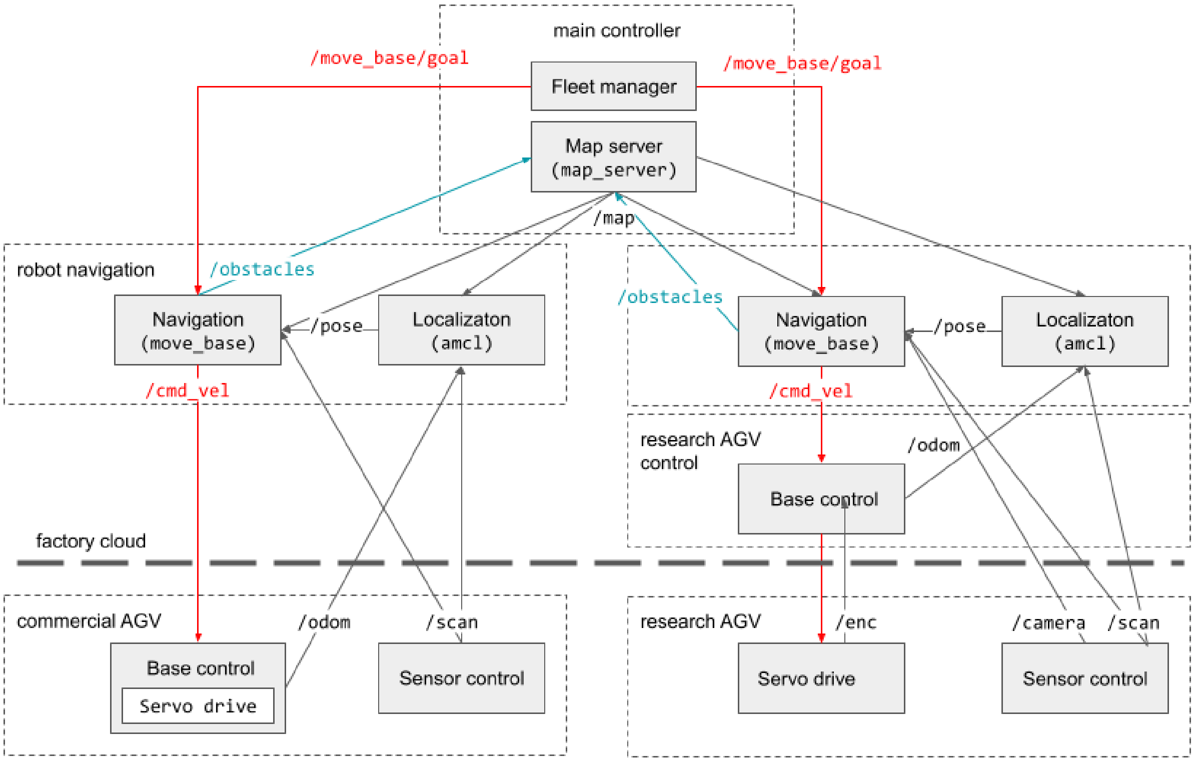


Figure 9 Main parts and communication diagram of the detailed software architecture

Navigation (*move_base* ROS package): The navigation stack uses *cost maps* to store information about obstacles in the world. The Navigation module is divided into the *global planner*, which uses a priori information of the environment to create the best possible path, if any, and the *local planner*, which recalculates the initial plan to avoid possible dynamic obstacles. The *global planner* requires a *map* of the environment to calculate the best route. The *local_planner* is responsible for computing velocity commands to send to the mobile base of the robot given a high-level plan.

Localization (*amcl* ROS package): Localization uses the Adaptive Monte Carlo Localization (AMCL) method [AMCL]. AMCL is a probabilistic localization system for a robot moving in 2D. It implements the adaptive Monte Carlo localization approach which uses a particle filter to track the pose of a robot against a known map. AMCL takes in a laser-based map, laser scans, and transform messages (odom), and outputs pose estimates.

Map server: The map server (*map_server* ROS package) offers map data as a ROS Service. The map describes the occupancy state of each pixel of the world. In the standard configuration, lighter pixels are free, darker pixels are occupied, and pixels in between are unknown. One example of a map is illustrated in Figure 10.

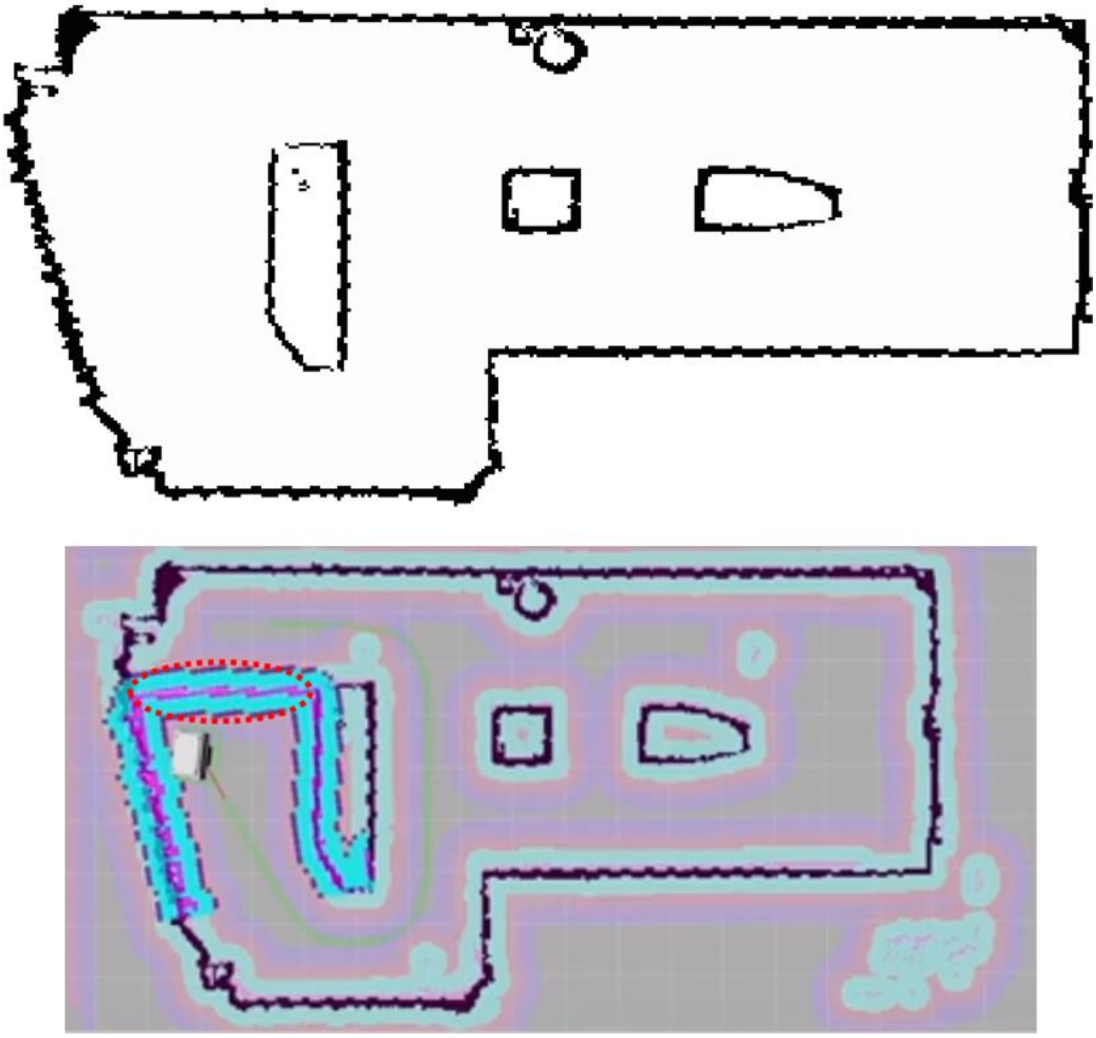


Figure 10 Map of the test area at Ericsson Hungary (top) and a snapshot of the GUI using the map for navigation (bottom) when the red circled part of the test route blocked by the tester

Fleet manager: The fleet manager node serves as the high-level main controller of the AGV fleet (i.e., the two AGVs in our case). It includes the map server functional block, and the fleet manager functionality. The latter implements the coordinated control of the two AGVs by issuing mission goals for the robot navigation components using and action interface and tracks the action states and feedbacks from the robots.

Base control: The base control node in the control stack of the research AGV implements the servo control functionality. It receives velocity commands that represents the intended motion of the whole AGV platform. This information needs to be translated to velocity commands per servo motor such that the intended motion of the platform is realized. This is our custom implementation, i.e., not available as an existing ROS package. It also contains built-in features to support the execution of multiple controller instances.

Sensor control: The sensor control node interfaces with the sensors, collects and structures data, as well as streams them towards the control nodes in the factory cloud for processing.

2.2.4 Factory cloud aspects

The project aims to demonstrate how the factory-local edge cloud-based control solution is able to service a large number of AGVs (*scalability*) and how it can overcome software or hardware failures (*resiliency*) without service interruption which is measured by mean time between failures (MTBF) which should be below an adequately defined threshold.

2.2.4.1 Benefits of the factory cloud

Scalability is demonstrated by deploying a large number (up to 100, depending on compute resource usage during the validation phase) of virtual AGV devices whose activity is simulated but otherwise indistinguishable from those of real devices by the other functional blocks. The expectation is that per-device functions will be seamlessly created and terminated as the number of virtual AGVs changes and global functions will keep providing their services to other blocks at the same performance level, i.e., response times.

Resiliency is demonstrated by purposefully stopping software components that actively serve one or more AGVs and check that the devices are still being controlled uninterrupted. For this to work, function blocks will either be replicated in a so-called *active-active* redundancy or restarted *reactively* by a supervising logic, depending on the criticality of each function. *Active-active* means that a function is carried out by more than one *instance* of the software simultaneously (e.g., two *Servo control* blocks for the same AGV) such that the other blocks communicating with them take this fact into account, i.e., by sending input to both and eliminating possibly duplicate replies. *Reactive* mode means that a single instance of a function is started and monitored periodically using active status requests, called *keepalive* or *health checking*. If a configurable small number of keepalive requests are unanswered then a new function instance is started that initializes its' state according to the surviving one through a replicated database updated periodically during function operations, and system configuration is changed to communicate with it from there on. Which of the two modes is employed for a specific function block depends on how critical is the uninterrupted operation of the function to the system: for latency critical ones the active-active mode provides the required resiliency at the expense of higher resource usage.

2.2.4.2 Factory cloud deployment

The AGV control software runs on both the moving devices and in the local factory cloud environment. For the purposes of this control, the 5G radio subsystem provides a connectivity between the two. During development of the control software components, the radio part will be substituted with a wired link for simplicity as illustrated in Figure 11.

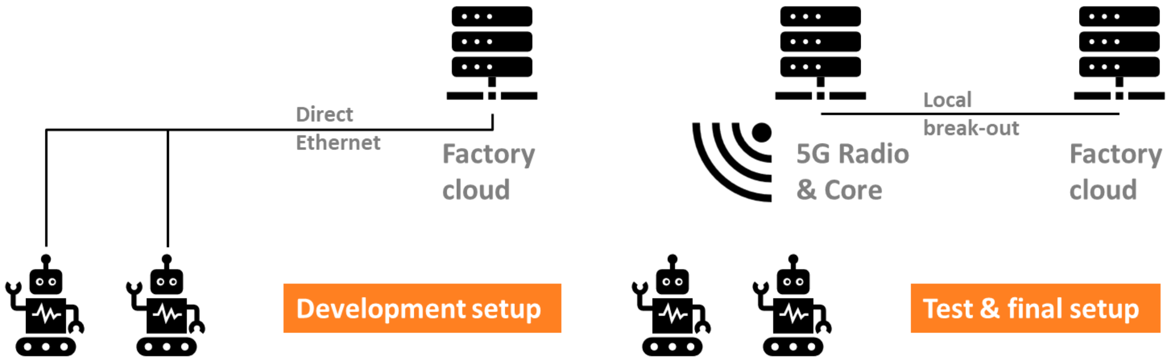


Figure 11 High level system architecture for development (left part) and for the final setup (right part)

The factory cloud is a stand-alone cloud platform running on factory premises in the vicinity of the radio subsystem to take advantage of the lowest possible transport latencies. It comprises three commodity x86 servers deployed in the rack space near the 5G core elements, as this is the minimum number required to demonstrate resiliency of the solution to a single-node failure. The cloud system employs distributed consensus for both data and process redundancy.

The infrastructure layer of the cloud (compute & storage servers, and networking capabilities) satisfies the ultra-low latency and high bandwidth networking requirements, that will be verified with appropriate measurements during the verification phase of the project.

Do best preserve hardware resources of an edge infrastructure layer and accommodate for brown-field setups that might be low on such resources, a lightweight virtualization platform is selected in the form of software containers (Docker), as opposed to, e.g., virtual machines (VM). Lightweight virtualization is considered more important in edge computing, even if VMs would provide better isolation of compute tasks at much higher resource needs, but which is not needed in the controlled environment of the factory (i.e., there are no unknown, “noisy” neighbors we would want to isolate from). The cloud platform provides orchestration and management for the software containers running the user application, and applications and services can be deployed in a standard way by configuring their containers. This cloud service model is also called Container as a Service (CaaS), and Kubernetes has become the *de-facto* standard for orchestrating such containerized workloads in data centers and the public cloud. Because Kubernetes provides a common layer of abstraction on top of physical resources — compute, storage and networking — developers or DevOps engineers can deploy applications and services in the same way on different Kubernetes clusters. This is important for our development process as it takes place in a different environment than the final deployment at the trial factory site.

Instantiating a fully-fledged Kubernetes environment includes the detailed configuration of multiple components, however as the purposes of this project do not concern with Kubernetes optimization we chose simpler approaches: for the development cloud we use MicroK8s inside Multipass virtual machines, and for the verification at the trial site we plan to utilize *Kubespray* [Spray], an easy-to-use bootstrapping tool.

2.3 End-to-end system integration

We have integrated the necessary sensors, drivers, and 5G modems to the AGV platforms including the power supplies, configuration, and network setup. Furthermore, the integration of the cloud environment is also done, the control components are running from software containers in the cloud execution environment. Some of the functional tests performed in Ericsson's test lab in Budapest have been demonstrated at various events, for example, at the Ericsson R&D Innovation Days in Budapest, Hungary. A demo video showing the results at November 2020 is available on the Youtube channel of the project [AGVDemo]. The final integration with the 5G network deployment in the Reutlingen factory is not yet performed due to travel restrictions imposed by Covid-19. The validation scenarios along with industrial KPIs to be evaluated for this use case will be reported in 5G-SMART Deliverable D4.4.

3 TSN/Industrial LAN over 5G in the shop-floor

The focus of this activity since the beginning of the project has been put into designing a detailed description of the use case scenario and evaluating how the use case can be implemented and validated during the run-time of the project. For this purpose, we have particularly dealt with two sets of challenges. First, TSN is a new and still-evolving technology. As a result, industrial controllers and networking components supporting TSN features (e.g., TSN switches) are not widely and commercially available yet, which in turn limits our choices for implementing the use case. Second, supporting TSN features in 5G system starts from Rel-16, and Rel-16-based products were not available at the time of starting deployment of the 5G network in the semiconductor factory. Therefore, our 5G network deployment is based on Rel-15 products, which limits our possibilities for testing TSN features over the 5G system. Accordingly, we have designed the use case implementations with these limitations in mind.

The idea of this use case is to validate the feasibility of transporting traffic concerning critical machine-to-machine applications over 5G communications networks. For this purpose, we will look into controller-to-controller (C2C) communication transported over an integrated 5G/Industrial LAN network. These applications could be for instance large machines like industrial printers or packaging systems, where different subsystems are managed through individual control units, i.e. programmable logic controller (PLC), and a timely exchange of information among these controllers is required for the desired operation of the system as a whole. Another example is the case, where different production units contribute to manufacturing a product and controllers at each of these units need to inform each other about status of the production.

3.1 End-to-end system architecture and use case

The end-to-end system architecture for this use case is depicted in Figure 12. The system is composed of two subsystems, namely the industrial automation subsystem as well as the 5G network subsystem. The 5G subsystem provides connectivity services for communications among components in the industrial automation subsystem. Specifically, we plan to set up three pairs of industrial automation components (color-coded with red, blue and green in Figure 12), where two components in each pair communicate with each other. This results in three separate traffic streams, Streams 1, 2 and 3. The first two streams carry experimental IP-based traffic based on C2C applications. C2C communication between the controllers will be deterministic, periodic and symmetric traffic, emulating a TSN traffic to some extent. Therefore, it is of paramount importance to validate if 5G is capable of satisfying the stringent requirements of the C2C communication, as detailed in D1.1 [5GS20-D110].

Stream 3 represents the communication between an operational industrial machine in the semi-conductor factory and its backend server. The task of this industrial machine, as detailed in 5G-SMART's Deliverable D4.1 [5GS20-D410], is to perform optical inspection of wafers by generating a number of photos of the review process, which are then uploaded to a database over a TCP/IP communication for post-processing that aims at classifying the detected wafer defects. The purpose of having the Stream 3 is twofold. First, we would like to check if 5G is capable of meeting the data

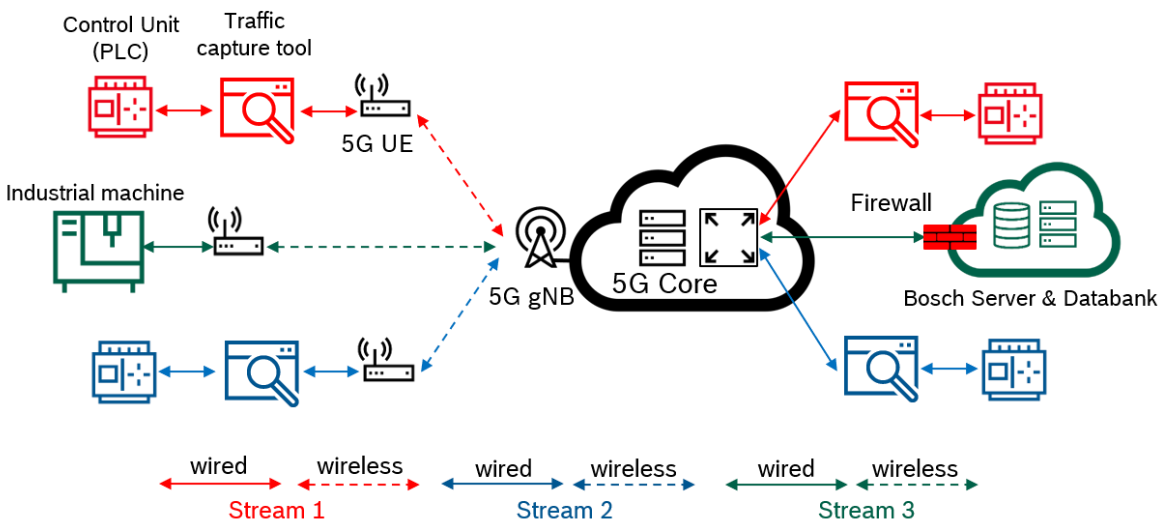


Figure 12 End-to-end system architecture for TSN/Industrial LAN use case. Stream 1 and 2 are both C2C traffic (i.e., PLC to PLC) and Stream 3 is communication between an industrial machine (Rudolph F30) and the server, acting as the background traffic for Streams 1 and 2.

rate requirement, which is around 50 Mbps, of this communication in order to guarantee its normal operation. Second, the Stream 3 is considered as a background traffic for the Streams 1 and 2 used to analyse the impact of the background traffic onto the C2C communication streams in terms of delay, for example. This is an important aspect to be evaluated in order to find out if the requiremets of the C2C communication streams can be satisfied with the presense of background traffic when no special traffic prioritization is in place and no network slice is created for the C2C communication streams. The result of this evaluation will be reporded in 5G-SMART's Deliverable D4.4. The hardware components used in both of the subsystems are presented below.

3.1.1 Hardware and software components

This subsection describes the hardware components required to realize the “TSN/Industrial LAN over 5G” use case. We have finalized the selection of these hardware components by taking into account their technical requirements, the available options with their characteristics, and their delivery timing. As for the details of the 5G network deployment and the used technology, we refer the reader to 5G-SMART’s Deliverable D4.1 [5GS20-D410].

5G UE

To enable the communication of the control units and industrial machines over the 5G network, both the control units of the Stream 1 and Stream 2, and the industrial machine of the Stream 3 have to be connected to 5G UEs, as displayed on the left side of Figure 12. To realize this communication, we obtained industrial modems (i.e., 5G UEs) based on Qualcomm X55.

Control Unit

We obtained “IndraControl XM22” control units [Indra] (depicted in Figure 13 (a)) as well as “S20-ETH-BK” bus coupler units [Eth] (depicted in Figure 13 (b)) from Bosch Rexroth to be used as PLCs for the Stream 1 and Stream 2. Modbus/TCP protocol is used by the control units to communicate with each other. IndraControl XM22 provides flexibility and programmability to support the implementation of various scenarios. We have programmed the control units with experimental applications, generating deterministic, periodic, and symmetric traffic, mimicking realistic scenarios so that they communicate with each other with programmable communication requirements, e.g. varying transfer intervals in the range of 4-10 ms as specified in [5GS20-D110].

(a) (b)

Figure 13 Industrial control unit IndraControl XM22 (a) and S20-ETH-BK bus coupler (b). The pictures are the properties of Bosch Rexroth.

Traffic capture tool

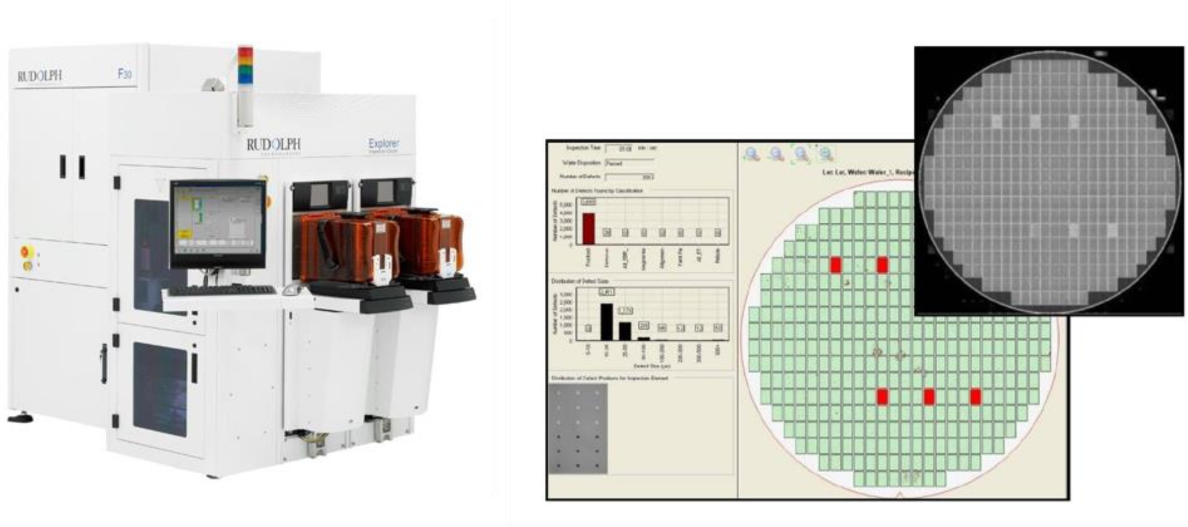
We procured ProfiShark 1G+ network taps [Profitap] (depicted in Figure 14), to estimate the end-to-end latency of the C2C communication and the delay introduced by the 5G system (i.e., the delay between the 5G UE and the 5G core network). To this end, one ProfiShark 1G+ network tap will be connected on the 5G UE side, capturing the C2C traffic from the control unit towards the 5G UE and vice verca, and another one will be connected on the 5G core network side, between the local breakout of the 5G core network the peer control unit, as illustrated in Figure 12, capturing the traffic from the 5G core network towards the peer controller and vice versa. ProfiShark 1G+ is endowed with advanced timestamping capabilities, which is vital for accurate latency estimation, which will be achieved by synchronizing the clock time of both network taps together.

Figure 14 Profishark 1G+, a portable traffic capture and troubleshooting tool.

Industrial Machine

For the Stream 3, we use an existing industrial machine called Rudolph F30 (as depicted in Figure 15 (a)). Rudolph F30 is a semiconductor production equipment for optical inspection of wafers in different steps of production process. Main functions of Rudolph F30 include:

- Frontside inspection (F30™ Module)
- Edge inspection (E30™ Module)
- Backside inspection (B30™ Module)
- Real Time Binning, real time classification (based on location, size, …)
- TrueADC for Review Images
- On the fly full Wafer images



(a) (b)

Figure 15 (a) Rudolph F30, (b) Output of the visual inspection. The pictures are properties of Bosch.

The result of the inspection are wafer coordinates of anomalies and abnormalities (defects), including a review and photos of this review (Figure 15 (b)). A single Rudolph F30 production equipment inspects around one hundred wafers per hour, generating a data volume of about 50 Mbps data, which is then loaded into the related database over a TCP/IP communication for automated classification of the detected defects. The detailed process description of wafer inspection can be found in 5G-SMART's Deliverable D4.1 [5GS20-D410].

3.2 End-to-end system integration

We have so far completed the detailing and specifying of various steps of the use case implementations including the end-to-end system architecture. We have also defined validation scenarios for this use case, which is further to be refined taking into account the technical features supported by the deployed 5G network. The finalized version of the validation scenarios along with the Key Performance Indicators (KPIs) to be evaluated for this use case will be detailed in 5G-SMART's Deliverable D4.4 towards the end of the project.

Additionally, we have integrated the controllers and the industrial machine into the 5G network. Specifically, we have connected an "S20-ETH-BK" bus coupler unit, which serves as a control unit, to a 5G UE on the one hand, and an "IndraControl XM22" control unit to the 5G core network-side on the other hand. The connectivity between the 5G UE and the "IndraControl XM22" control unit has been successfully tested. Besides, we have connected the Rudolph F30 industrial machine to a 5G UE, and once we have the end-to-end connectivity successfully tested for both of the communication types (i.e., C2C communication and F30 to its backend server communication), we will start the experiments with this use case according to the validation scenarios and evaluate the defined KPIs.

4 Summary and future work

In this report we discussed the 5G use cases implemented for the trial in the Bosch plant. The system architectures and implementations are discussed for the cloud based mobile robotics (collaborative AGVs), as well as, the TSN/Industrial LAN use cases. We present the detailed software and hardware architectures and status of the implementation. The development is ongoing at the time of the preparation of the report.

The factory cloud features such as the reliability and scaling are to be implemented for the collaborative AGVs. The definitions of KPIs (for different categories such as quality, flexibility, mobility, productivity, safety, utilization and sustainability) and the validation scenarios are under discussion at the moment among the partners. The final set of KPIs need to be implemented and evaluated during the defined validation scenarios. The connectivity testing between the integrated industrial components for the TSN/Industrial LAN over 5G use case is currently ongoing which will be followed by conducting experiments according to the validation scenarios and evaluating the KPIs. The results of the experiments for both use cases will be reported in 5G-SMART Deliverable D4.4 towards the end of the project.

Appendix A: Selection of AGV platforms

We have reviewed many commercial AGV platforms and examined their capabilities focusing on the availability of accessible interfaces for remote control and on the different certificates the platform is compliant with. We summarize the results in Table 1.

Some of these platforms provide Robot Operating System (ROS) support. ROS is an open source set of tools and libraries [ROS] that enable developers to create robotic applications easily. The support typically means that vendors develop drivers to their products which implement the hardware-dependent part of ROS.

Products	Certificates	Development aspects (interfaces, simulation support, etc.)	Remarks
MiR 100	CE, Cleanroom	ROS support, REST API, third party Gazebo simulation model available	
Robotnik RB series		ROS support, Gazebo is supported by the vendor	
Neobotix MP family	CE	ROS and Gazebo support by the vendor, alternative PlatformCtrl interface	Active ROS development, but few sensors are integrated on the HW
Metralabs SCITOS series	CE, Cleanroom	Own developed MIRA platform instead of ROS, no known Gazebo support	
Nanotec RP series		Limited ROS support by the vendor, no known Gazebo support	
Fabmatics HEROFAB	CE, Cleanroom	Limited third party interface only	Possible compatibility with Neobotix through PlatformCtrl interface
Omron LD series	CE, Cleanroom	No third party interface	External control not possible

Table 1 Summary of reviewed commercial AGV platforms

ROS also enables to execute some part of the logic outside of the HW platform. Whether a ROS-based architecture supports the millisecond-scale remote robot control heavily depends on the actual implementation of the ROS driver, which may vary a lot for different vendors. It was found to be infeasible to evaluate each ROS driver without having and testing the actual HW for evaluating the feasibility of the planned use case features.

Gazebo is an open-source, well-designed simulation toolbox [Gazebo] which can accurately and efficiently simulate populations of robots in complex indoor and outdoor environments. It includes a robust physics engine, high-quality graphics, and convenient programmatic and graphical interfaces. The availability of the Gazebo simulation model is an important aspect during development, since the digital twins of both robot platforms can be created and accurately simulated, which means that the constant physical access of the robots is not needed.

or makes it possible to rapidly test algorithms, design robots, perform regression testing, and train AI system using realistic scenarios. physics engine, high-quality graphics, and convenient programmatic and graphical interfaces. Best of all, Gazebo is free with a vibrant community.

As we illustrated in Table 1, we reviewed several commercial AGVs and **MiR 100 was selected** [MiR] in the end as it was found to be the most suitable. Since we aim to implement collaboration between the two AGVs by exploiting the factory cloud deployment, we must exchange information between the cloud and the commercial AGV device such as the actual position of the AGV and commands for locations to go, etc. The MiR 100 AGV has a ROS driver and a rich REST API [MiRAPI], and therefore, it is deemed as suitable for the purpose.

However, the main problem of relying on only commercial AGVs in this use case is that without evaluating the available ROS driver for the actual platform, it is uncertain whether the planned use case features are feasible to be realized. These ROS APIs are usually developed and maintained by a 3rd party group and can become non-compatible with the platform in case of an update. Furthermore, some of the cloud features such as reliable execution of the application using multiple control components at the same time may not supported by ROS. We note that the next version of ROS called ROS 2 is much more suitable for cloud-based execution [ROS2], but the list of available packages needed for the implementation are not available yet.

Based on the above reasoning, we investigated other alternatives and reviewed several research AGV platforms to find a more flexible option in terms of control possibilities and cloud-based execution. We also examined other solutions such as creating an AGV by ourselves from basic robotic components to make sure we have a hardware truly supporting all our needs.

Products	Certificates	Development aspects	Remarks
HEBI Mobile Base with R-Series smart servos	CE for all components, not for the whole platform	Open and low-level API of servos, ROS support	Servo control API up to 1 kHz, no built-in safety
Clearpath Dingo		ROS support	LiDAR, stereo camera and ultrasonic sensor integrated
Pal Robotics Tiago Base	MD 2006/42/EC	Web UI and REST API available	
Husarion ROSbot 2.0		ROS support	Small platform (20 x 20 cm), no place for load

Table 2 Summary of reviewed research platforms

Most of the investigated research AGVs listed in Table 2 were not favorable, since either they do not fulfill all requirements, or we could not collect enough technical information to evaluate. We found that HEBI Robotics [HEBI] produces robot servos that can fully cover our needs in terms of low-level control. A HEBI R-Series smart servo provides open and well documented low-level APIs for position, velocity and effort control, so we can implement the millisecond-scale, closed-loop motion control executed remotely from a factory cloud, which is needed, since it is part of the use case description, i.e., to run millisecond-scale control from the cloud. HEBI Robotics also builds robotic kits based on customer requests, so we **selected the HEBI Mobile Base** option as the research AGV to be used. We designed a simple mobile base platform that is tailor-made for our use case and asked HEBI Robotics for production. The mobile base consists of four smart servo motors (HEBI R8-3) that allow control cycle up to 1 kHz and can generate detailed feedback with the same frequency.

Important to note that the selected research platform is not an AGV by itself, it is only a simple metal chassis with four servo motors attached. We need to add and integrate sensor devices and their drivers to create an AGV hardware platform. This means that we aim to build an AGV from scratch including not only the hardware but also all layers of control components to be able to demonstrate the full potential of cloud-based execution of robot control. Although the research platform lacks certificates for obvious reasons, we designed to have only CE certified components.

Appendix B: Certifications

Since the safety department at Bosch Reutlingen requires CE and cleanroom certified equipment to be used on the shop floor, we agreed on making the complete HEBI based research AGV platform including attached sensors and 5G UE terminal CE compliant by a certification company such as TÜV Rheinland Ltd. The research AGV platform was already designed with safety and cleanroom requirements in mind. This means that we designed the system using safety sensors, safety relays, safety emergency button, waterproof motors and electronics (IP67) without fans or rotating parts to minimize dust generation. It has been agreed that Bosch makes a clean room test of the robot in their lab in Reutlingen before the validation, so there will be no additional cost introduced and the required extra workload is also minimal.

For the CE certification part, we initiated a process with TÜV Rheinland to certify the research platform. The complete CE certification process contains machinery, functional safety, EMC and radio parts. Since the cost of the complete process is very high (>80 k€), duration is long (>6 months) and it requires high amount of additional work, we agreed with Bosch to find alternative solutions. One such alternative is to use both robots in a supervision mode where a person follows every movement with an emergency stop button in hand. This solution would limit the possibilities of continuous operation of the robots in 24/7, but all of the planned use case scenarios can be evaluated and it is accepted by the safety department of the factory, so the robots are allowed to do missions on the factory shop floor.

Appendix C: Sensors for the research AGV

We shortlisted some sensors that can be attached and integrated into the research AGV (Table 3). The aim is to evolve the platform into a full-featured AGV robot, but not all sensors from the table are needed for that.

Sensors	Type	Certifications	Remarks	Quantity
Safety laser scanner	SICK S300 Standard S30B-3011BA	EU declaration of conformity, ACMA declaration of conformity, China-RoHS, CULus certificate, EAC certificate / DoC, EC-Type-Examination approval, China GB certificate	Safety conformity IEC 61496, EN ISO 13849	2
	Hokuyo UAM-05LP-T301	CE, RoHS	IEC61496-1, IEC61496-3, IEC61508	2
LiDAR	Hokuyo UTM-30LX	FDA Approval (21 CFR, 1040.10 and 1040.11)	Framed to IP67 box, 270° view	2
	Velodyne VLT-16	CE, RoHS, CTC 6776	Framed to IP67, mid range	1
	RPLIDAR A2M8	FCC and CE	Rotating top part	1
Depth camera	Intel RealSense D435i	EU Declaration of Conformity, RoHS	Good reviews	2
Ultrasonic sensor	Maxbotix I2CXL-MaxSonar-EZ4	CE, FCC and RoHS	For each side of the platform	4

Table 3 Set of proposed sensors for the research AGV platform

The safety laser scanner can be integrated directly with the servo motors to decrease the speed or stop the whole platform when triggers are received from the sensor. The HEBI R-Series servo modules have connectors that can put the servo into M-STOP state when 5V-48V input power is switched to the connectors. M-STOP is a forced stopped (emergency stop) state of the motor where the servo stops immediately and will return to operation when the input power is switched off.

The LiDAR and the depth camera data are needed to implement Simultaneous Localization and Mapping (SLAM), i.e., to explore the surrounding area and localize itself on the continuously updated map. The ultrasonic sensors are cheap devices to help detecting objects around the AGV. Another option is to use the output of the safety laser scanners if available, and in this case, there is no need for dedicated LiDARs.

As shown in Table 3, we have reviewed multiple models within the same sensor type, but we need only one model. The exact quantity depends on the actual model. Regarding the LiDARs, the first two types are sealed (IP67), which means that there is no moving part visible. These models have only 270° view, thus there is no rotating laser diode inside but rather sets of time of flight (TOF) sensors. The Velodyne LiDAR has 360° scanning angle which is most probably realized by a rotating laser sensor and the whole mechanics are boxed and special glass added that is transparent for the scanner. Velodyne VLT-16 is applied on several other AGV platforms although it is designed for mid-range scanning up to 100m. The RPLIDAR has a rotating part, but according to colleagues at Bosch Reutlingen, it is still allowed to be applied in the clean room part of the factory.

List of abbreviations

AGV	Automated Guided Vehicle
API	Application Programming Interface
C2C	Controller-to-Controller
CaaS	Container as a Service
CAD	Computer-Aided Design
CE	Conformité Européenne (French for "European Conformity")
DDS	Data Distribution Service
HW	Hardware
I-LAN	Industrial Local Area Network
LiDAR	Light Detection and Ranging
NR	New Radio
PC	Personal Computer
ReST	Representational State Transfer
ROS	Robot Operating System
SLAM	Simultaneous Localization and Mapping
SPC	Statistical Process Control
SW	Software
TSN	Time Sensitive Networking
UE	User Equipment
UI	User Interface
VM	Virtual Machine
YM	Yield Manager

References

[HEBI] "HEBI Robotics, https://www.hebirobotics.com/"

[Indra] "Bosch Rexroth IndraControl, IndraControl XM22".

[Eth] "Bosch Rexroth S20-ETH-BK, S20-ETH-BK".

[Kub] "Lightweight Kubernetes, https://k3s.io/".

[Spray] "Kubespray: Deploy a Production Ready Kubernetes Cluster, https://kubespray.io/"

[5GS20-D110] 5G-SMART https://5gsmart.eu/wp-content/uploads/5G-SMART-D1.1.pdf, "Forward looking smart manufacturing use cases, requirements and KPIs", June 2020.

[5GS20-D410] 5G-SMART https://5gsmart.eu/wp-content/uploads/5G-SMART-D4.1-v1.0.pdf, "Report on design and installation of the 5G trial system in Reutlingen", November 2020.

[5GS21-YT] https://www.youtube.com/watch?v=4hxVlfGnhUg&t=1s

[ROS] "Robot Operating System", https://www.ros.org/".

[Gazebo] "Gazebo simulation tool, http://gazebosim.org/".

[MiR] "MiR 100, https://www.mobile-industrial-robots.com/en/solutions/robots/mir100/".

[MiRRest] "MiR 100 REST API documentation, https://www.mobile-industrial-robots.com/media/11210/mir_mir100_rest_api_21002.pdf".

[ROS2] "Robot Operating System version 2, https://docs.ros.org/en/foxy/index.html".

[AGVDemo] "Cloud based mobile robotics demo at Ericsson R&D Innovation Days, https://youtu.be/xBWDwmFGPm8, Budapest, Hungary, 10-11 November 2020".

[AMCL] "Adaptive Monte Carlo localization method, http://wiki.ros.org/amcl".

[Profitap] "Profishark 1G+, https://www.profitap.com/profishark-1g-plus/".

[5GS20-CT] "5G-SMART – 5G Common terminology", June 2020. https://5gsmart.eu/wp-content/uploads/5G-SMART-common-terminology.pdf