D3.4 – i4Q Trusted Networks with Wireless and Wired Industrial Interfaces

WP3 – BUILD: Manufacturing Data Quality

i4Q has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 958205.

Document Information

GRANT AGREEMENT NUMBER	958205	ACRONYM	i4Q
FULL TITLE	Industrial Data Services for Quality Control in Smart Manufacturing		
START DATE	01-01-2021	DURATION	36 months
PROJECT URL	https://www.i4q-project.eu/		
DELIVERABLE	D3.4 – i4Q Trusted Networks with Wireless and Wired Industrial Interfaces		
WORK PACKAGE	WP3 – BUILD: Manufacturing Data Quality		
DATE OF DELIVERY	CONTRACTUAL: June 2022	ACTUAL	June 2022
NATURE	Report	DISSEMINATION LEVEL	Public
LEAD BENEFICIARY	ITI		
RESPONSIBLE AUTHOR	José Vera (ITI), Santiago Gálvez (ITI)		
CONTRIBUTIONS FROM	1-TIAG, 2-KBZ, 3-ENG		
TARGET AUDIENCE	1) i4Q Project partners; 2) industrial community; 3) other H2020 funded project; 4) scientific community		
DELIVERABLE CONTEXT/ DEPENDENCIES	This document presents a technical overview of the Trusted Network with Wireless and Wired Industrial Interfaces solution (i4Q^{TN}). This document has no preceding documents but will have a new iteration called D3.12 "i4Q Trusted Networks with Wireless and Wired Industrial Interfaces v2", that will be delivered at M24.		
EXTERNAL ANNEXES/ SUPPORTING DOCUMENTS	None		
READING NOTES	None		
ABSTRACT	This deliverable presents a technical overview of the i4Q Trusted Networks with Wireless and Wired Industrial Interfaces solution (i4Q^{TN}), covering the technological description of their components and its mapping against the i4Q Reference Architecture. In addition, an analysis of the current state of solution up to M17 is performed, showing the relationship between the technical requirements, described in previous deliverables, and the current functionalities of the solution. After describing in detail, the developments performed, final sections of this deliverable explain the remaining implementation works.		

Document History

VERSION	ISSUE DATE	STAGE	DESCRIPTION	CONTRIBUTOR
0.1	10-May-2022	ToC	Table of Contents available	ITI
0.2	27-May-2022	1st Draft	First Version Technical Deliverable	ITI
0.3	08-Jun-2022	Internal review	Review and comments	BIBA, RIAS
0.4	22-Jun-2022	2nd Draft	Second Version Technical Deliverable	ITI, TIAG
1.0	30-Jun-2022	Final Draft	Final quality check and issue of final document	CERTH

Disclaimer

Copyright message

TABLE OF CONTENTS

Executive summary **7**

Document structure **8**

1. General Description **9**

1.1 Overview 9

1.2 Features 9

2. Technical Specifications **10**

2.1 Overview 10

2.2 Architecture Diagram 10

3. Implementation Status **12**

3.1 Current implementation 12

3.1.1 Solution features analysed and mapping with user requirements 21

3.2 Next developments 22

3.3 History 22

4. Conclusions **24**

References **25**

Appendix I **26**

LIST OF FIGURES

Figure 1. i4QTN Solution Architecture 11
Figure 2. IWSN Gateway internal architecture 13
Figure 3. SDN controller and IWSN validation tests 14
Figure 4. Left: HUB platform v0.0.1; Right: HUB and node combined in v0.1.0 15
Figure 5. TSN queues and transmission gates 16
Figure 6. Selected TSN standards 16
Figure 7. Architecture TSN Network Configuration Tool 18
Figure 8. Topology editor for deterministic network planning 18
Figure 9. Stream path visualization and network schedule 19
Figure 10. Embedded Configuration Tool 20
Figure 11. TSN Evaluation Board 21

LIST OF TABLES

Table 1. i4QTN version history 23

ABBREVIATIONS/ACRONYMS

API	Application Programming Interface
CNC	Central Network Controller
CUC	Centralized User Configuration
DNS	Domain Name System
FRER	Frame Replication and Elimination for Reliability
IEEE	Institute of Electrical and Electronics Engineers
IT	Information Technologies
IWSN	Industrial Wireless Sensor Networks
LLDP	Link Layer Discovery Protocol
MAC	Medium Access Control
MQTT	Message Queue Telemetry Transport
NETCONF	Network Configuration Protocol
NG	Next-Generation
OPC UA	Open Platform Communications United Architecture
PCB	Printed Circuit Board
QoS	Quality of Service
RA	Reference Architecture
REST	Representational State Transfer
SDN	Software Defined Networking
SDN WISE	SDN solution for WIreless SEnsor Networks
SPI	Serial Peripheral Interface
SRP	Stream Reservation Protocol
TAS	Time-Aware Shaper
TN	Trusted Networks with Wireless and Wired Industrial Interfaces
TSCH	Time Slotted Channel Hopping
TSN	Time Sensitive Networking
UML	Unified Modelling Language
WSN	Wireless Sensor Networks
YANG	Yet Another Next Generation

Executive summary

This document presents an executive explanation of the **i4Q Trusted Networks with Wireless and Wired Industrial Interfaces** (i4Q^{TN}) Solution providing the general description, the technical specifications and the implementation status. The deliverable **D3.4** is the Source Code of the i4Q^{TN} Solution that is in a private repository of Gitlab: https://gitlab.com/i4q.

The documentation associated to the i4Q^{TN} Solution is deployed on the website http://i4q.upv.es. This website contains the information of all the i4Q Solutions developed in the project "Industrial Data Services for Quality Control in Smart Manufacturing" (i4Q). The direct link to the i4Q^{TN} Solution documentation is http://i4q.upv.es/4_i4Q_TN/index.html.
Such documentation is structured according to:

- General description
- Features
- Images
- Authors
- Licensing
- Pricing
- Installation requirements
- Installation Instructions
- Technical specifications of the solution
- User manual

Document structure

Section 1: Contains a general description of the **i4Q Trusted Networks with Wireless and Wired Industrial Interfaces (**i4Q[TN]**)**, providing an overview and the list of features. It is addressed to final users of the i4Q Solution.

Section 2: Contains the technical specifications of the **i4Q Trusted Networks with Wireless and Wired Industrial Interfaces (**i4Q[TN]**)**, providing an overview and its architecture diagram. It is addressed to software developers.

Section 3: Details the implementation status of the **i4Q Trusted Networks with Wireless and Wired Industrial Interfaces (**i4Q[TN]**)**, explaining the current status, next steps and summarizing the implementation history.

Section 4: Provides the conclusions.

APPENDIX I: Provides the PDF version of the **i4Q Trusted Networks with Wireless and Wired Industrial Interfaces (**i4Q[TN]**)** web documentation, which can be accessed online at: http://i4q.upv.es/4_i4Q_TN/index.html.

1. General Description

1.1 Overview

The **i4Q Trusted Networks with Wireless and Wired Industrial Interfaces (**i4Q^{TN}**)** is a software-defined industrial interface for data communication, characterized by predictability and determinism, high reliability, trustability and low consumption while reducing the cost of new communication infrastructures. i4Q^{TN} ensures high-quality data collection, providing connectivity to industrial data sources through Trusted Networks able to assess and ensure precision, accuracy, and reliability.

1.2 Features

- Provide a reliable and secure communication infrastructure.
- Trustable platform for Low Latency communications.
- High availability, scalability and flexibility using private infrastructure.
- Wired and wireless interfaces.
- QoS guarantees, orchestrating network resources using a SDN controller.

2. Technical Specifications

2.1 Overview

This solution will work with Hybrid wireless-wired networks with high Quality of Service assurance, including Software defined networks (SDN, slicing) applied to wireless sensors technologies (WSN) with state-of-the-art mechanisms to ensure robustness while operating in industrial environments characterized by a high level of noise and interferences. Also, focusing on the wired part of the solution, deterministic ethernet based on Time Sensitive Networks (TSN) in being developed to provide an even further strict time response for applications with hard real-time requirements.

This solution is hardware dependent, requiring specific network interfaces and gateways to relay data. Network optimization can be customized to adapt to different use cases, and scalability can be addressed by adding more gateways and router in case of need. Security can be covered with the encryption of data and channels or provided by other i4Q solutions whose main efforts are the cybersecurity of communications using certificates or different security approaches.

The **i4Q Trusted Networks with Wireless and Wired Industrial Interfaces (**i4Q^{TN}**)** is mainly mapped to the Data Collecting sub-component of the Edge Tier, since it allows to guarantee a certain level of Quality of Service (QoS) in the current data acquisition communication infrastructure, such as high reliability of data acquisition, low latency data exchange for time critical applications or the enough bandwidth to transmit high volumes of information.

2.2 Architecture Diagram

i4Q^{TN} includes different communication technologies which are mapped to the Edge Tier Layer in the i4Q Reference Architecture. **Figure 1** shows this solution mapping against the i4Q Reference Architecture, highlighting in red the “Data Collecting” subcomponent from the Edge Tier and in orange colour all the assets and smart products from which the data collected comes, such as IoT sensors, IT systems or industrial machines.

- **Edge Tier**: The solution covers the main communication interfaces that are needed in different industrial scenarios, using wired, wireless and low power IoT systems. Although the solution is mapped in Edge Tier, the information exchange through i4QTN solution can be used by any other solution at any RA level.

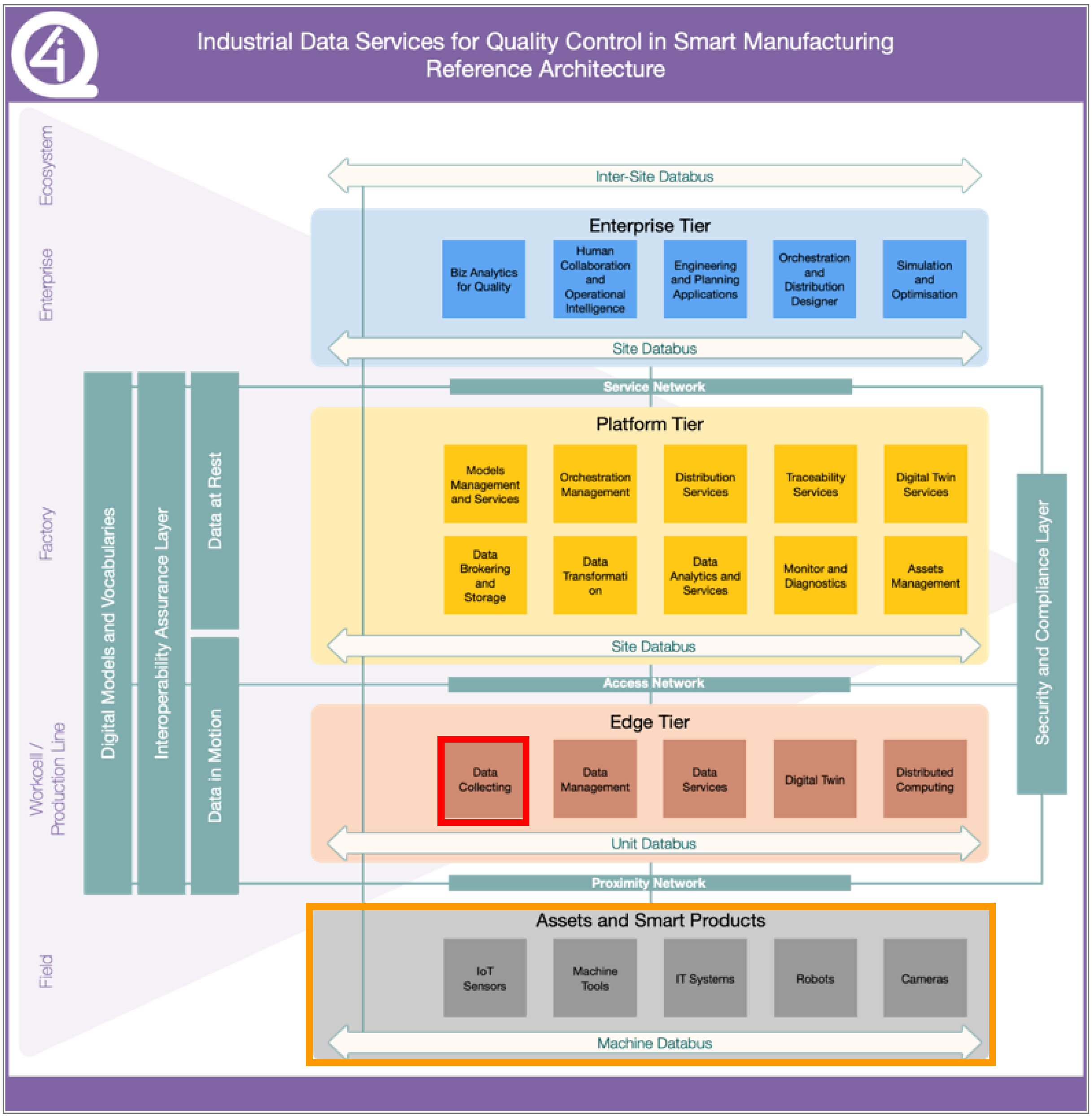


Figure 1. i4QTN Solution Architecture

3. Implementation Status

3.1 Current implementation

The **i4Q Trusted Networks with Wireless and Wired Industrial Interfaces** (i4Q^{TN}) is based on a set of communication technologies to collect data from different industrial processes and systems. The use of different technologies makes it possible to cover a wide range of applications and use cases, deploying wired solutions to interconnect wit already deployed networks and optimize the access method to obtain a deterministic data exchange; using wireless solutions for mobile use cases, and allowing to install acquisition nodes on difficult-to-access areas; and low power consumption networks to deploy a large number of nodes along a production process or carry out itinerant measurements with a low energy cost.

The current implementation of the i4Q^{TN} Solution can be summarized into two major technological components:

- A communication system based on WSN technologies has been developed, implementing the standard IEEE 802.15.4 with TSCH as medium access method, orchestrated using a SDN controller.
- A communication system based on IEEE 802.1 TSN technologies has been developed.

Wireless communication interfaces

The first communication technology is based on a low power consumption wireless communication, which allows the deployment of Wireless Sensor Networks based on Ad Hoc mesh architecture. This type of networks allows greater distances and improve the communication resilience thanks to its multi hop link communications and the ability to adapt the network topology against changes, selecting alternative routes in case of losing a link or replacing a network device.

The system is based on the IEEE 802.15.4e standard, using TSCH as a medium access mechanism, allowing nodes to exchange information in a deterministic way, configuring time slots to different links and allowing the selection of guaranteed slot to avoid collision between different nodes. This feature, in combination with the frequency hopping mechanism, which allows mitigating interference in a certain channel, allow this technology to operate in industrial scenarios, where reliability and communication robustness becomes more important.

The system architecture is based on three main components:

- The **communication gateway** establishes a link between two access technologies, allowing the traffic generated to be routed towards ethernet based networks. This component allows the IWSN interaction with other application or system deployed in plant or with other i4Q solutions using different Industry 4.0 communication protocols.
- **SDN controller**, involved in processing report messages collected from all nodes belonging to the network and in the orchestration of the TSCH the resources in time and frequency, to guarantee the Quality-of-Service (QoS) requirements defined by the application, such as the sensor reception period or the guaranteed resource allocation.
- **Wireless nodes**, which allow the deployment of a mesh topology either to collect data from different industrial-range sensors or to retransmit the information from those nodes that are not within the coverage area of the communication gateway.

The development of the first two components has been carried out on the same physical equipment, working both as a gateway and SDN controller, deploying these services in a Linux-based architecture. **Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε.** shows the internal architecture of the IWSN developed gateway.

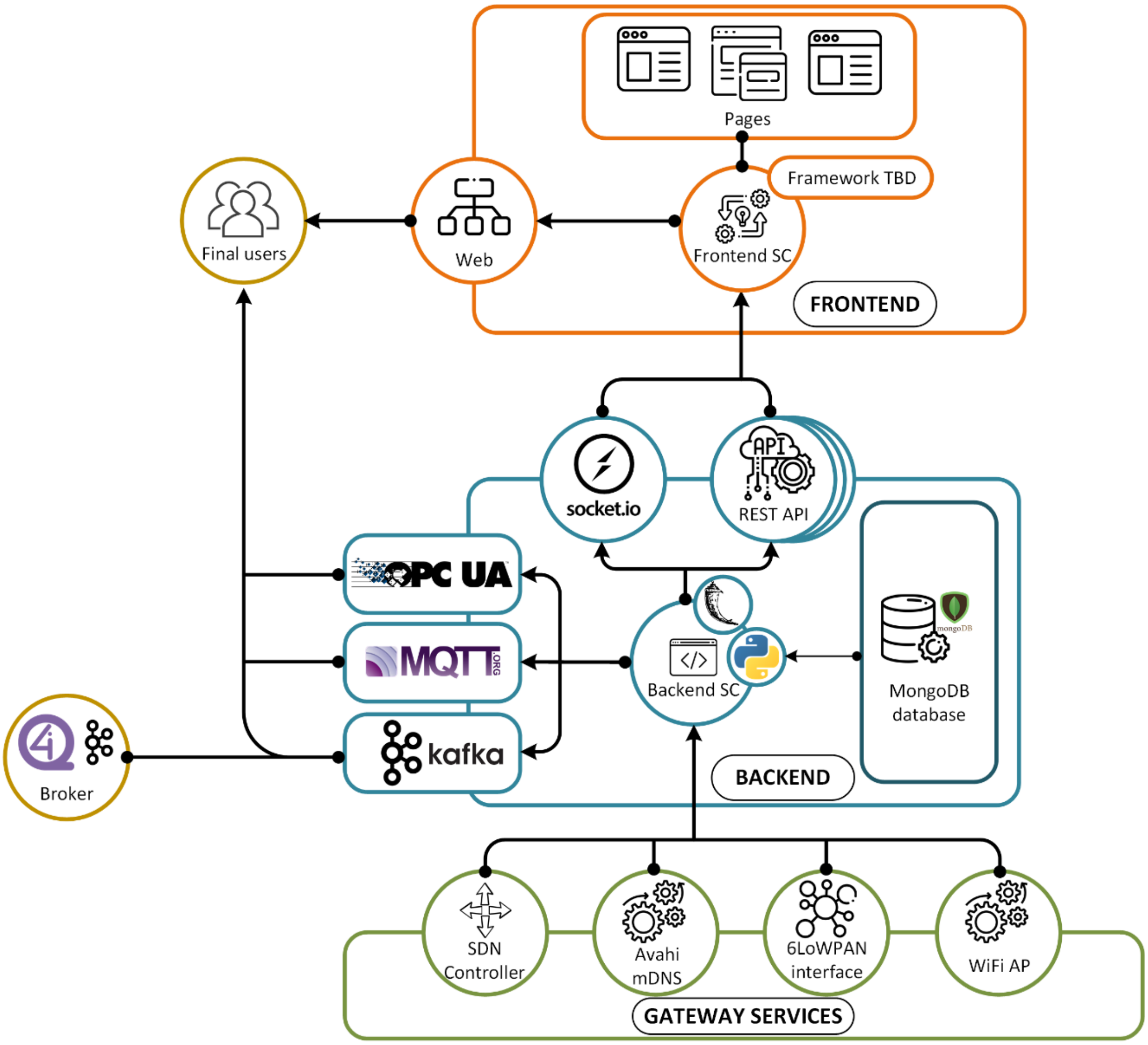


Figure 2. IWSN Gateway internal architecture

The system is able to work independently, without a direct connection to the plant network, managing the devices and visualizing the streaming of collected data through the gateway services, or integrate with other systems of the i4Q architecture using standardized Industry 4.0 interfaces, such as the Kafka i4Q message broker. The gateway has three well-differentiated system processes:

- **Generic gateway processes**: first it allow the embedded system based on linux to connect with the IEEE 802.15.4 nodes using a 6LoWPAN interface, which acts as a border router between both technologies; enables a WiFi access point to be able to access the gateway in case the system is not connected to the plant network, as well as a multicast DNS service to easily identify the equipment on the network based on its network name, defined based on its MAC address; the SDN controller, in charge of managing the report messages transmitted by the nodes of the IWSN and orchestrating the network resources, based on the Quality-of-Services policies defined by the application or use case.
- A **backend service** that allows managing the information generated by the IWSN, publishing this information to the different communication protocol enabled in the system, or enabling an interface to transfer this information to the frontend service or the internal database. In addition, the backend controls the system initialization and checks if the rest of the system processes are running after the start up. Furthermore, this service manages the configuration of individual nodes by the selection of the proper sensor and the acquisition period.
- A **frontend service** that will allow managing the entire system from a web portal, configuring the nodes, representing the network statistics and the system status, or visualizing the data generated by the IWSN in real time.

The firmware of the wireless nodes has been developed based on the operative system Contiki NG, which allow to implement WPAN networks using TSCH mechanism. In addition, to exchange information with the SDN controller and allowing the configuration of the TSCH resources, the solution SDN WISE has been integrated, developing some functionalities to enables the management of TSCH resources. **Figure 3** shows the test validation of the SDN controller and different sensor nodes of the IWSN, deploying only the isolated services of the SDN controller, using the integrated network visualizer and the TSCH resource management tool, and the interface with the IEEE 802.15.4 network.

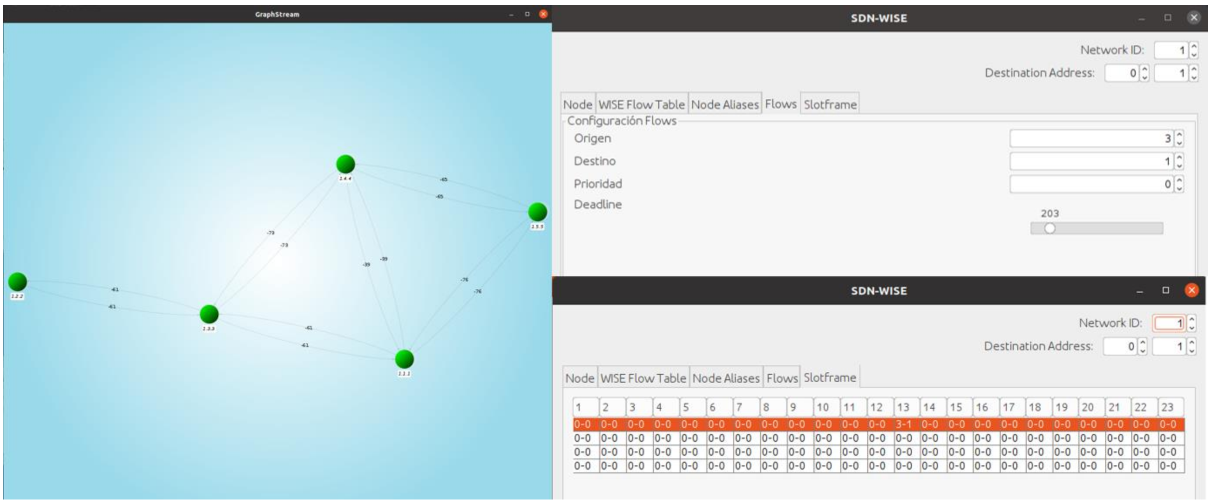


Figure 3. SDN controller and IWSN validation tests

For the development of the wireless nodes, in parallel with the software developments, the deployments of this nodes in industrial scenarios need to be considered, allowing connection to 24V sources, as well as collecting information from standardized industrial sensor interfaces, such as 4-20mA, 0-10V, 24V Digital Inputs and RS485 bus interfaces, to stablish an exchange of data with protocols such as Modbus-RTU. For this reason, the design of a PCB has been performed to integrate the CC2538 TI radio transceiver and the necessary interface to adapt the platform from typical 3V3-5V to 24V signals, as well as the previously mentioned sensor interfaces.

This process has been carried out in several iterations, initially isolating these elements into two subsystems (the wireless node and the industrial sensor HUB). In a second phase, these components have been integrated into the same PCB design, simplifying the interconnection between the two units, and integrating the new RS-485 interface, not available in the previous version. This interface allows connection to Modbus-RTU devices, commonly used in many energy measurement systems deployed in industry.

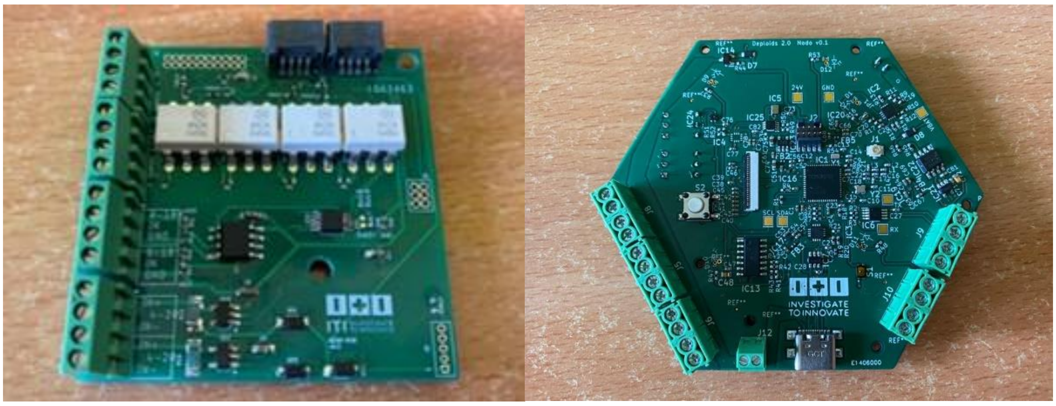


Figure 4. Left: HUB platform v0.0.1; Right: HUB and node combined in v0.1.0

Figure 4 shows the evolution of the IWSN hardware platform. On the left, first hardware version of the HUB platform, including the following industrial interfaces: 4-20mA, 0-10V and 24V Digital Inputs. On the right, the second version combining the communication transceiver and the industrial interfaces, adding the RS-485, USB-C power supply and programming interface and other elements such as the flexible connector for e-Paper displays via SPI communications.

Wired Communication Interfaces

The second communication technology is based on the extensions of IEEE 802 Ethernet standard, known as Time-Sensitive Networking (TSN), which enables deterministic real-time communication over Ethernet. TSN achieves determinism over Ethernet by using time synchronization and a schedule which is shared between network components. By defining queue-based message forwarding based on time (see **Figure 5**), TSN ensures a bounded maximum latency for scheduled traffic through switched networks. That means that in a TSN network, latency of critical scheduled communication is guaranteed. In control applications with strict deterministic requirements, such as those found in automotive, industrial, aerospace and space domains, TSN offers a way to send time-critical traffic over a standard Ethernet infrastructure. This enables the convergence of all traffic classes and multiple applications in one network. In practice this means that the functionality of standard Ethernet is extended, so that (1) message latency is guaranteed through switched networks; (2) critical and non-critical traffic can be converged in one network; (3) higher layer protocols can share the network infrastructure; (4) real-time control can be distributed across a wider area; (5) sub-systems can be integrated more easily; (6) the system being more robust against changes in the network; and (z) network faults can be diagnosed and repaired faster.

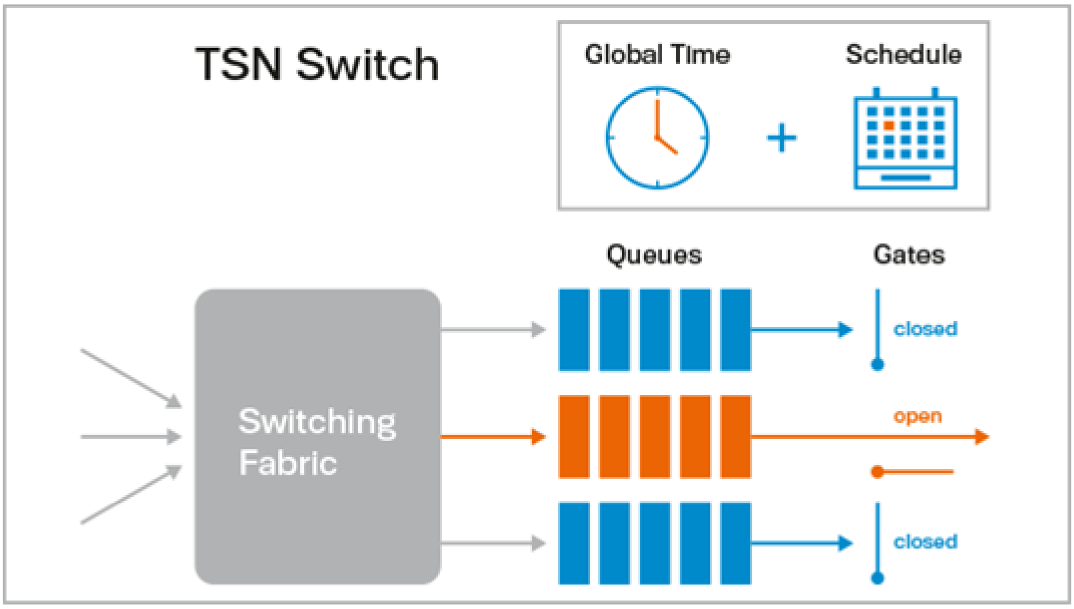


Figure 5. TSN queues and transmission gates

As mentioned, TSN is a collection of standards that support different features (see **Figure 6**).

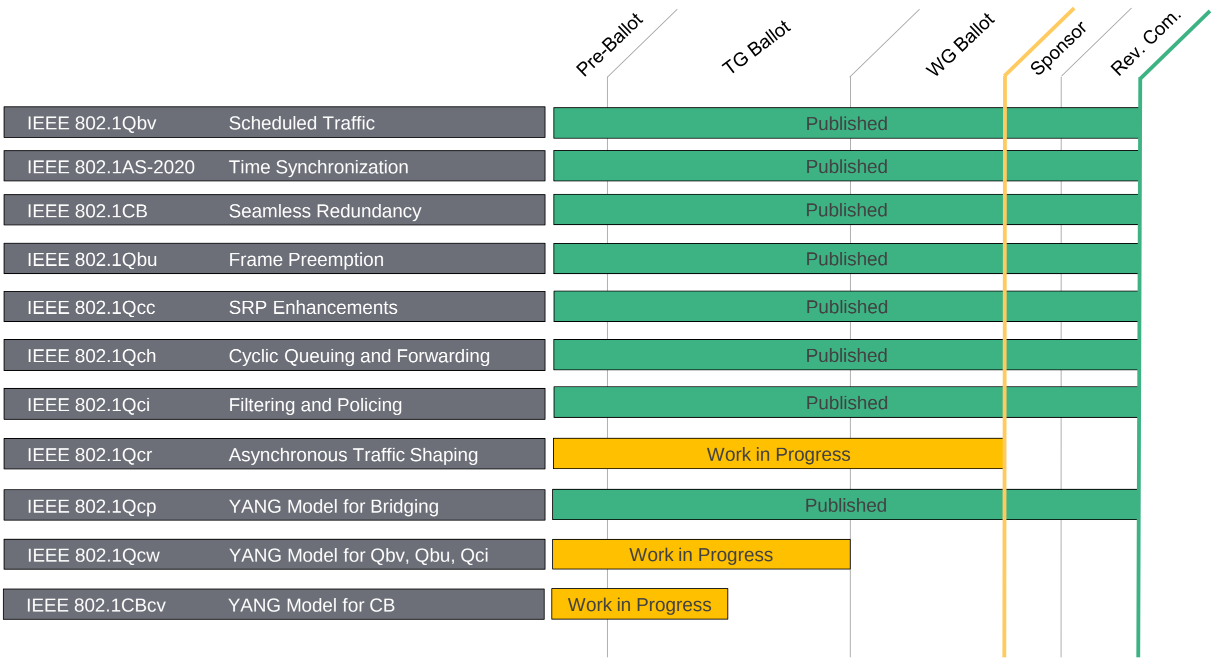


Figure 6. Selected TSN standards

The solutions provided support the following set of standards from the overall TSN standards:

- **IEEE 802.1Qbv – Scheduled Traffic:** Scheduling traffic is at the core of TSN, which is a real-time, determinism and reliability communication protocol. Within TSN, this is known as the Time-Aware Shaper (TAS), which deterministically schedules traffic in queues through switched network (see **Figure 5**). IEEE 802.1Qbv-2018 standardizes this. The TAS controls the flow of queues traffic from a TSN-enabled switch. Ethernet frames are identified and assigned to queues based on the priority of the frames. Each queue operation is based on a schedule, and the transmission of messages in these queues is then executed at the output ports during the scheduled time windows. Other queues are then blocked, thereby removing the change of scheduled traffic being interrupted by non-scheduled traffic.
- **IEEE 802.1AS – Timing and Synchronisation:** To establish deterministic communication with bounded message latency using TSN, clock synchronisation is a vital mechanism. IEEE 802.1AS enables the creation of a profile of the IEEE 1588 PTP (Precision Time Protocol) synchronization protocol between different TSN devices. Furthermore, IEEE 802.1AS standardizes the use of multiple grandmaster clocks as well as the possibility to make connection to these grandmaster clocks. The replication of grandmaster clocks will result in shorter fail-over times in case a grandmaster clock becomes faulty. Additionally, multiple synchronized clocks are supported by IEEE 802.1AS, which enables timestamping of evets (e.g., production data or measurements) and the synchronization of applications such as sensors, actuators and control units.
- **IEEE 802.1Qcc – Stream Reservation Protocol (SRP) Enhancements and Performance Improvements:** IEEE 802.1Qcc enables consistent configuration of Ethernet switches from different vendors, making TSN vendor independent. It also provides mechanisms to improve existing reservation protocols, such as SRP (Stream Reservation Protocol – IEEE 802.1Qat) in order to meet the configuration requirements of industrial (and automotive) systems. These are requirements like e.g., timing, bandwidth reservation, frame pre-emption, synchronization, and redundancy. Furthermore, it supports the implementation of central configuration models for dynamic scheduling of TSN networks.

One of the challenges with TSN networks is how to plan, organize and schedule these networks. Therefore, a network planner specifically targeting deterministic networks using TSN is needed, that enables system integrators and designers to plan the overall topology and configuration of the deterministic network. **Figure 7** provides an architecture overview of the network configuration planner for deterministic networks. The configuration tool has a planning engine, which takes care of the planning of the network and the scheduling. It has a Central Network Controller (CNC), which defines the schedule on which all TSN frames are transmitted. Additionally, a Centralized User Configuration (CUC) application can be included which is the application that in principle communicates with the control applications and the end devices. It makes requests for the CNC for deterministic communication (TSN flows) with specific requirements for these flows.

The device configuration of the components in the network is acquired using NETCONF, enabling the planning engine to create a complete configuration of the whole network and distributed it also over the network. This way all the switches in the network are aware of the generated schedule for sending messages over the network. The human user than activates the deployment of the schedule to the whole network, additionally informing the applications in the network about the timing offset caused by the generated schedule.

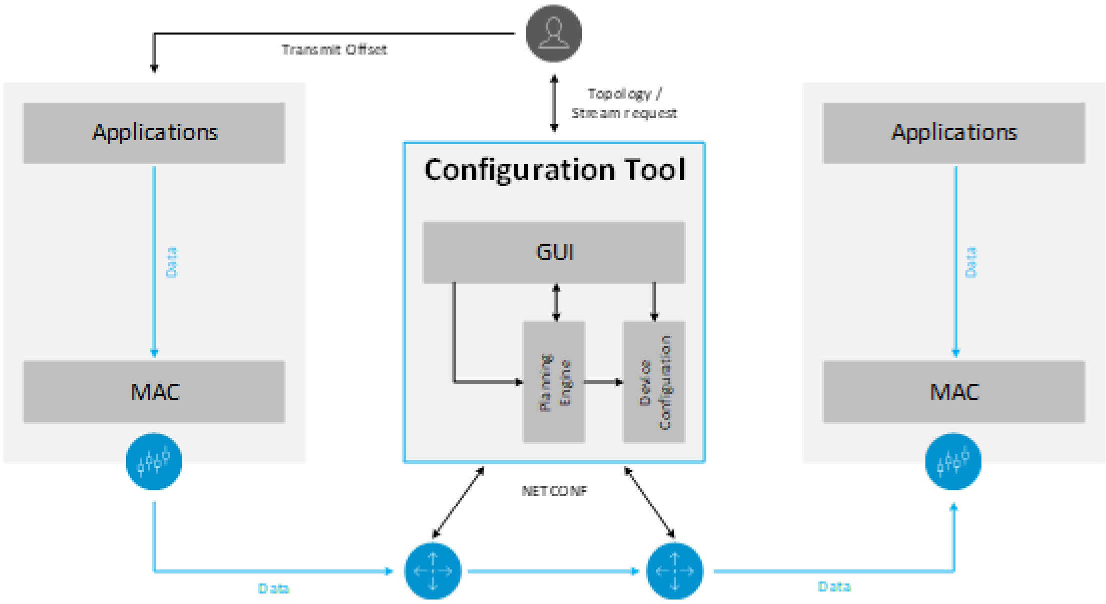


Figure 7. Architecture TSN Network Configuration Tool

The configuration tool is a browser-based, user-friendly software that makes it easy to model topologies, add streams and deploy configurations. It provides intuitive graphical topology modelling (as depicted in **Figure 8**) in either a graphical or a table editor format, which is better applicable for modelling large network topologies.

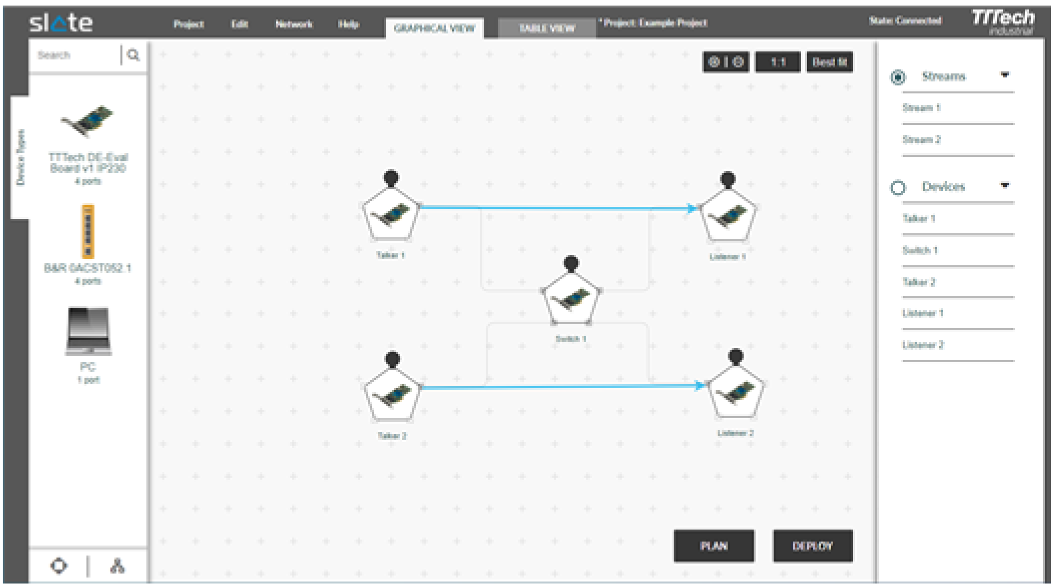


Figure 8. Topology editor for deterministic network planning

The planning engine is directly built into the configuration tool. After building the topology and defining the different data streams within the network, the complete configuration of the network can be created with a single click. **Figure 9** shows additional information provided by the solution, presenting stream path visualization and the generated schedule for the network.

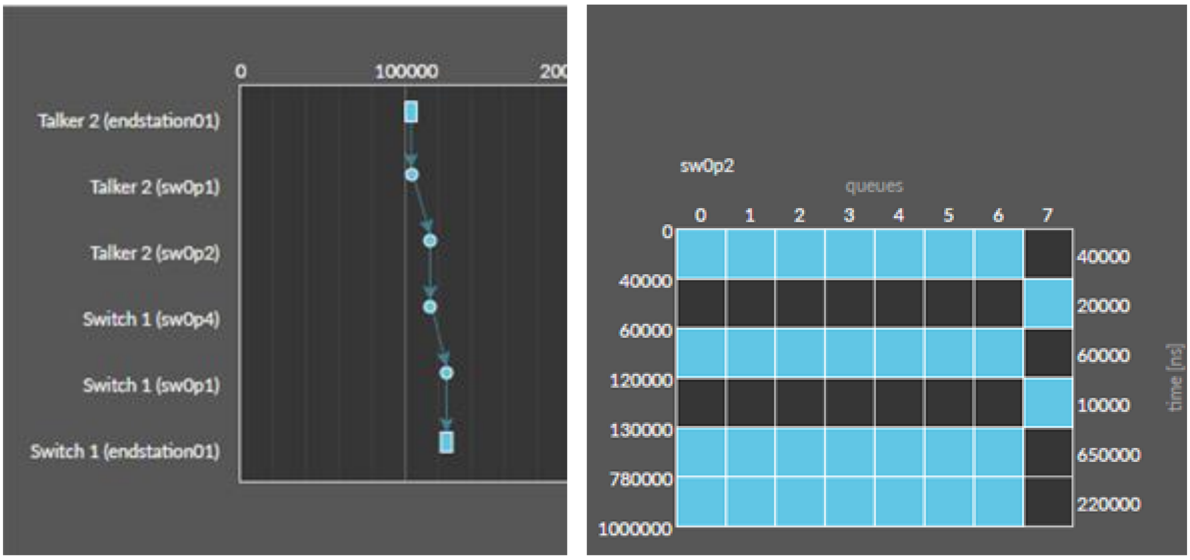


Figure 9. Stream path visualization and network schedule

The next step in the configuration of Deterministic Networks is that the overall system is capable of identifying changes in the network itself and updates the configuration based on the changes in the network (e.g., topology, parameters change, etc.). The alternative approach is that the solution is provided as an embedded software that runs directly in a device that is located in the network and facilities the planning of data streams according to the requirements of the applications. The solution automatically configures all standard TSN components by using OPC UA Publisher and Subscriber protocol. Publisher and subscribers are used to inform the system of new TSN streams that are available or are demanded by specific components in the network. Demands (or requests) can include end-to-end communication parameters, like cycle time, maximum latency, or seamless redundancy. The discovery of the network components making up the topology of the network, is established using the Link Layer Discovery Protocol (LLDP) and the capabilities of the individual devices using YANG. The stream configuration for the network is generated for the network based on the application requirements. The application can reserve streams before publishers and subscribers go online. As soon as the request is generated, the network is being (re-)configured. An updated (embedded) architecture can be found in **Figure 10**.

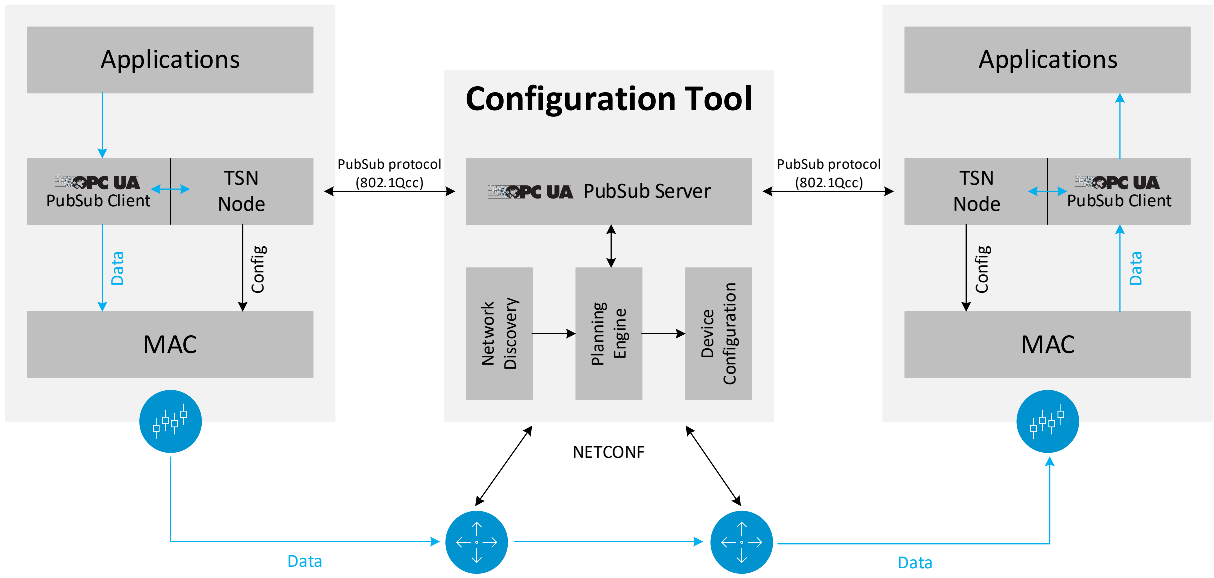


Figure 10. Embedded Configuration Tool

For the support of deterministic network, developments have been undergoing and are still going on the support the following IEEE 802.1 TSN standards to improve the performance of the network configuration:

- **IEEE 802.1Qcp – YANG Model for Bridging:** This standard specifies a UML-based information model and a YANG data model that enables the configuration and status reporting for bridges and bridge components.
- **IEEE 802.1Qcw – YANG Model for Qbv, Qbu and Qci:** This standard specifies a UML-based information model and a YANG data model that enables the configuration and status reporting for bridges and bridge components with the capabilities of scheduled traffic (Qbv), frame pre-emption (Qbu) and per-stream filtering and policing (Qci).
- **IEEE 802.1CBcv – YANG Model for CB**: This provides a YANG data model that enables configuration and status reporting for bridges and bridge components with the capabilities of Frame Replication and Elimination for Reliability (FRER) and Stream Identification.

An evaluation board is available for testing the generated schedule and configuration (see **Figure 11**), Together with a switch that supports TSN, a network can be set up for deploying the configuration. The evaluation board is a switch board designed to be used in combination with our ^{DE}IP solution. The combination of the board with our IP provides a stable platform for integration and evaluation of TSN Ethernet functionality for use in industrial automation applications.

Figure 11. TSN Evaluation Board

3.1.1 Solution features analysed and mapping with user requirements

A set of features has already been developed for i4Q^{TN}, based on the set of user and pilot requirements referring to i4Q^{TN} [1] and in line with the functional viewpoints [2]. Similar requirements have been assigned into common categories of tasks based on an extensive technical study conducted on user requirements, introduced to ensure the generalization abilities of the i4Q^{TN} solution.

The following list show in more detail the relationship between the solution's features and the requirements defined in D1.9 [1] for the i4Q^{TN}.

- *04ITIr1 "Full network orchestration":* is covered by the features implemented by the SDN controller, obtaining a centralized management of the network resources.
- *04ITIr2 "Improve node mobility":* is supported by the improvements obtained with the integration of the SDN controller, giving a centralized overview of the whole network to control the topology changes in a fastest way.
- *04ITIr4 "Reliable communications":* is covered by the implementation of the TSCH access method and other algorithms and mechanism implemented to improve the synchronization process of the IWSN.
- 04TIAGr1 "Bandwidth": Currently, applications don't require a maximum bandwidth of 10Gbit/sec. Most likely the targeted generic pilot will use standard 1Gbit Ethernet. Scalability of the TSN functionality is under evaluation towards 10Gbit.
- 04TIAGr2 "TSN Support": Each component in the network (i.e., switches, bridges, end systems) that want to use TSN must be able to understand TSN. This is covered by installing the correct IP on the component to enable TSN:
- 04TIAGr3 "Master Clock": The Grand Master Clock required for TSN is covered by the IEEE 802.1AS standard by using the IEEE 1588 PTP (Precision Time Protocol).
- 04TIAGr4 "Traffic convergence": Traffic convergence is covered by the IEEE 802.1Qbv standard that schedules the traffic in the network. This standard schedules the different messages in the networking, and enables the convergence of scheduled messages and standard Ethernet messages on the same network.
- 04IKERr1 "Protocol Translation":

- *PC4r1.2 "Ensure reliability in the exchange of information":* is covered by the implementation of the TSCH access method and other algorithms and mechanism implemented to improve the synchronization process of the IWSN.

3.2 Next developments

The solution status up to M17, on the one hand allow the deployment of a functional IWSN ready to work in industrial scenarios, considering not only the reliability and robust communication methods, but also the hardware adaptation to industrial signals and interfaces. On the other hand, the support of TSN is provided by the solutions from TIAG. Integrating, deploying TSN will be investigated in the second phase of the project targeting the generic pilot. Here is it however important to identify which application can and will benefit from exploiting TSN. However, some features and developments, listed below, are pending as future works.

- Kafka publisher implemented in python must be integrated with the gateway platform, taking as example the code provided by the message broker solution provider.
- Frontend development for the management of the gateway platform, adapting the API/REST developed in the backend service.
- Simulation tool to give some advice during the configuration process of each sensor node and the selection of the proper sensor acquisition period.
- Include a human-machine-interface to support the end users during deploying stages, giving information related to the link quality, remaining battery, and any kind of relevant information.
- Guide of use and configuration of the IWSN platform.
- Pilot use case specific development of sensor drivers.
- Continuous development of support of new TSN standards.
- Identification of TSN benefits in the generic pilot use case.

3.3 History

This section provides the version history of the i4Q^{TN} solution implementation up to M17, which is summarized in the table below. The version history differentiated between software version (SW) including all the developments related to software implementation for all the system involved, and hardware version (HW) including only the works related to the design and validation of the hardware platform.

Version	Release date	New features
V0.0.1 (SW)	30/11/2021	Mechanism implementation for the IWSN to improve synchronization times during deployment stages.
V0.0.2 (SW)	23/12/2021	SDN-WISE integration with Contiki NG and TSCH.
V0.0.1 (HW)	23/12/2021	Validation of the first HUB version, some issues regarding the interconnection with the node (dedicated USB-C) detected.
V0.0.3 (SW)	31/01/2022	SDN controller planification service for TSCH resources.

Version	Release date	New features
V0.0.4 (SW)	10/02/2022	New drivers developed for 4-20mA sensors and Modbus-RTU.
V0.0.5 (SW)	15/02/2022	Backend and SDN controller integration, updating API/REST framework to interact with SDN controller functions.
V0.0.6 (SW)	08/04/2022	Updated Java SDN controller with new function integration with the gateway backend.
V0.1.0 (HW)	30/05/2022	Second version of the Hardware platform updated for M18 solution release.
V0.1.0 (SW)	30/05/2022	M18 solution release.

Table 1. i4Q^{TN} version history

4. Conclusions

Deliverable "D3.4 - i4Q Trusted Networks with Wireless and Wired Industrial Interfaces" is a technical specification document, in which the technical and development aspects of the i4Q Trusted Networks with Wireless and Wired Industrial Interfaces solution ($i4Q^{TN}$) are specified. It describes in detail the role, the functionalities, and the conceptual architecture of $i4Q^{TN}$. The main features and functionalities of the solution are also clarified, describing its diagram model based on the i4Q Reference Architecture.

In addition, the developments and studies related to the $i4Q^{TN}$, and carried out during this first period, are described in detail in this document, showing the status of the $i4Q^{TN}$ solution up to M17. Finally, a list of pending features and developments is also provided in order to complete the solution implementation in M24.

References

[1] i4Q, *D1.9 – Requirements Analysis and Functional Specification v2.* (Sep 2021)

[2] i4Q, *D2.6 – Technical Specifications,* (Sep 2021)

Appendix I

i4Q Trusted Networks with Wireless and Wired Industrial Interfaces (i4Q[TN]**)** web documentation can be accessed online at: http://i4q.upv.es/4_i4Q_TN/index.html

i4Q Trusted Networks with Wireless and Wired Industrial Interfaces

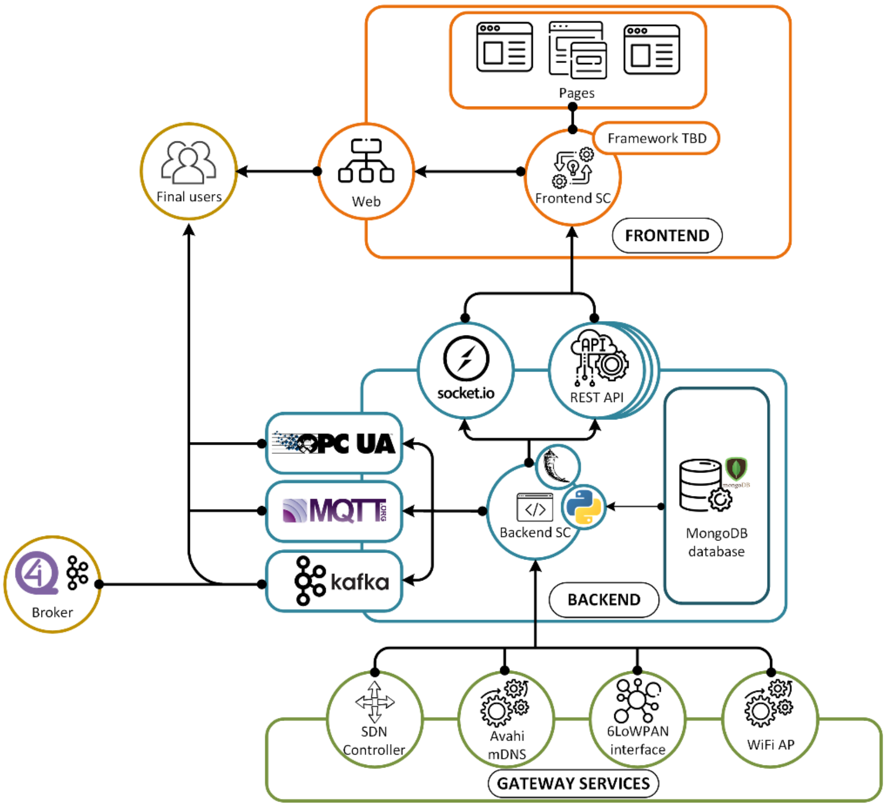
General Description

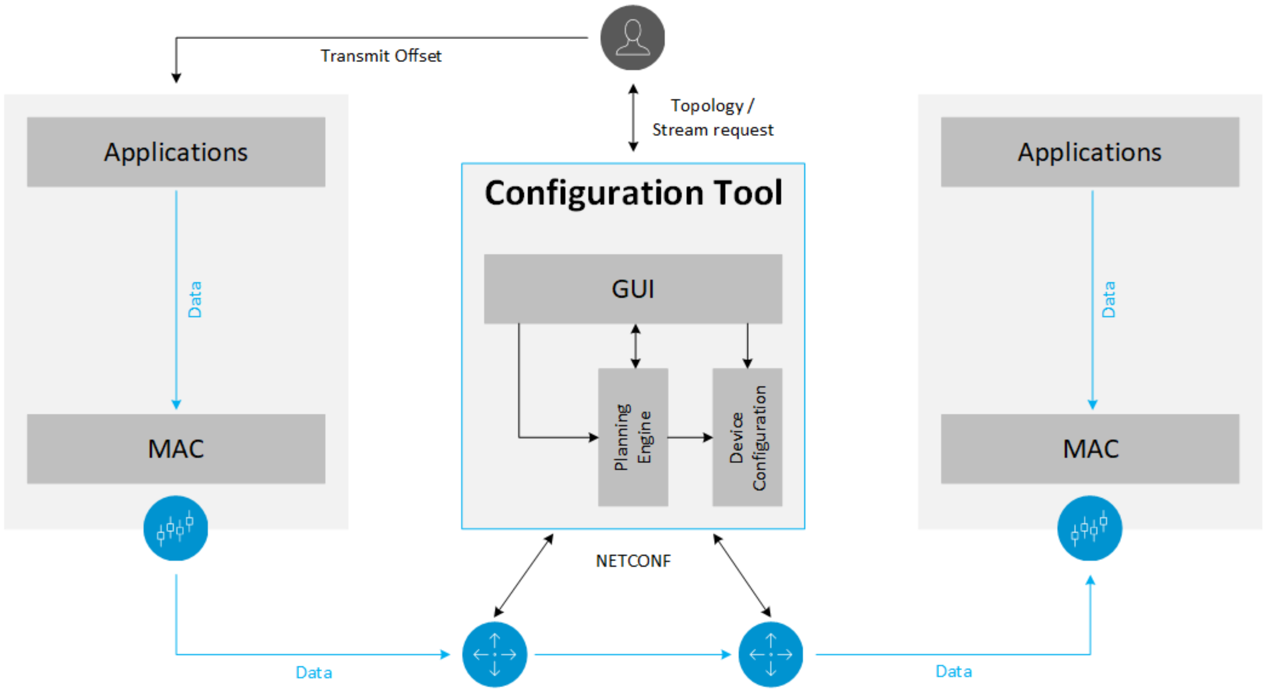
i4Q[TN] is a software-defined industrial interface for data communication, characterized by predictability and determinism, high reliability, trustability and low consumption while reducing the cost of new communication infrastructure. i4Q[TN] ensures high-quality data collection, providing connectivity to industrial data sources through Trusted Networks able to assess and ensure precision, accuracy and reliability.

Features

- Provide a reliable and secure communication infrastructure.
- Low Latency communications.
- High availability, scalability and flexibility using private infrastructure.
- Wired and wireless interfaces.
- QoS guarantees, orchestrating network resources using a SDN controller.

ScreenShots





Commercial Information

Authors

Partner	Role	Website	Logo
ITI	Leader	www.iti.es	ITI INVESTIGATE TO INNOVATE
ENG	Vice-Leader	www.abc.com	ENGINEERING
KBZ	Participant	www.abc.com	KNOWLEDGE BIZ

Partner	Role	Website	Logo
TIAG	Participant	www.tttech-industrial.com	TTTech industrial

License

TBD

Pricing

Subject	Value
Payment Model	TBD
Price	TBD €

Associated i4Q Solutions

Required

i4Q Trusted Networks with Wireless and Wired Industrial Interfaces solution has no dependency on another i4Q solutions and none of the i4Q solutions are dependent on i4Q[TN].

Optional

- i4Q IIoT Security Handler
- i4Q Data Repository
- i4Q Data Integration and Transformation Services

System Requirements for Wireless interfaces

- **IWSN Node**: hardware platform based on CC2538 TI transceiver for each node to be deployed.
- **IWSN Gateway**: SBC with IEEE 802.15.4 interface (i.e Raspberry Pi 4 with CC2538 transceiver using a tunslip6)
- Different industrial or IoT sensors to be connected to the HUB platform (4-20mA, 0-10V, 24V DI, Modbus/RTU, i2C)

System Requirements for Wired interfaces

- **TSN Nodes**: hardware platforms (switches, bridges, end systems) supporting collection of standards from IEEE 802.1 TSN (e.g., IEEE 802.1Qbv, IEEE 802.1Qcc, IEEE 802.1AS)
- Data transfer via Ethernet

- Windows 10 or Linux Ubuntu 18.04 for running the offline TSN Network Configuration Tool

API Specification for the IWSN Gateway

Resource	POST	GET	PUT	DELETE
/api/v1/login	Supported	Not Supported	Not Supported	Not Supported
/api/v1/refresh	Supported	Not Supported	Not Supported	Not Supported
/api/v1/gatewaystatus	Not Supported	Supported	Not Supported	Not Supported
/api/v1/wsn	Not Supported	Supported	Not Supported	Not Supported
/api/v1/<int:nodeid>/sensors	Supported	Supported	Supported	Supported
/api/v1/wsn/upload	Not Supported	Not Supported	Supported	Not Supported
/api/v1/config/network	Supported	Not Supported	Supported	Not Supported
/api/v1/config/database	Supported	Not Supported	Supported	Not Supported
/api/v1/config/mqtt	Supported	Not Supported	Supported	Not Supported
/api/v1/config/opcua	Supported	Not Supported	Supported	Not Supported
/api/v1/config/kafka	Supported	Not Supported	Supported	Not Supported

Installation Guidelines

Resource	Location
Last release (v.1.0.0)	Link Documentation
Video	Link

Installation of the IWSN system

The complete installation of the IWSN subsystem, included in i4Q Trusted Networks with Wireless and Wired Industrial Interfaces solution, requires the compilation and installation of different software componentes in different hardware platforms. First, the firmware compilation and updload for the Wireless Sensor nodes, based on Contiki NG OS, SDN-Wise and the needed developments to combine the use of Time-Slotted Channel Hopping (TSH) medium access method with the proper functions to support SDN scheduling rules and quality reports to the SDN controller. All the components **requires specific hardware platforms, and the subsystem can not be virtualized or containerized**, since the different platforms requires a direct connection with different physical interfaces, such as the 6LoWPAN network interface or the local system configuration to set up the WiFi AP or the Avahi mDNS services.

Uploading the firmware to the nodes

To upload the firmware to the nodes it is necessary to set up the Contiki-NG Toolchain, either through a native installation (Linux) or using the official Docker images. The easiest and fastest way is to use the official Contiki-NG images. For further information regarding the native or Docker installation, please follow the steps of the official Wiki.

```
$ docker pull contiker/contiki-ng
```

Instead of clonning the official repository (ContikiNG), download the compressed ZIP file with the i4Q ContikiNG and start docker with the following options:

```
$ docker run --privileged --sysctl net.ipv6.conf.all.disable_ipv6=0 --mount type=bind,source=<absolute-path-to-the-downloaded-i4q-contiking>,destination=/home/user/contiki-ng -e DISPLAY=$DISPLAY -v /tmp/.X11-unix:/tmp/.X11-unix -v /dev/bus/usb:/dev/bus/usb -ti contiker/contiki-ng
```

Finally, to build and upload the firmware, the IWSN node must be attached to the contiki-ng docker container, and the serial device must be specified to properly upload the firmware to the IWSN node (the following example shows the `/dev/ttyUSB0` serial device). The `TARGET=zoul` variable must be included, since the hardware platform developed is based on CC2538 TI transceiver and some other common components based on zoul platform pin assignment. To configure different node ids for each IWSN node, different hexadecimal numbers must be defined using the `NODEID` inline variable.

```
$ cd examples/i4qtn-iwsn/
$ sudo make TARGET=zoul MOTES=/dev/ttyUSB0 NODEID=0x0001 i4q-node.upload
```

In case any problem occurs, the firmware can also be uploaded using the **SmartRF Flash Programmer 2** from Texas Instrument and the `.hex` firmware image of the i4q-node, using the TI XDS110 Debug Probe.

Start the SDN controller and the IWSN Gateway

The IWSN Gateway is based on a Single Board Computer (Raspberry Pi 4 or Tinker Board) that requires additional components to be able to interconnect with the 6LoWPAN IWSN nodes. Any mote with the CC2538 transceiver (Zolertia RE-Mote, Zolertia Firefly, Openmote or even the i4Q

IWSN hardware developed) must be attached to the IWSN Gateway to define a new network interface to communicate with the IWSN nodes.

Since the IWSN Gateway requires different subcomponents to properly deploy the IWSN network, a bin bash script has been defined to schematically supervise the deployment process of each component of the IWSN Gateway. The official i4Q TN repository, with the developments up to M18, only host the components of the SDN controller and the backend subcomponent of the complete platform. The final version will explain in more detail the rest of the subcomponents.

```
sudo ./start.sh
```

This script supervise the deployment of each of the following subcomponents: mDNS Avahi service, 6LoWPAN network interface, local WiFi Access Point, local MongoDB database, Java SDN Controller and the backend service.

Installation of the TSN system

The complete installation of a TSN system is not a straight-forward task. It is more than just setting up the hardware with TSN IP, which is a first requirements for supporting TSN in the system. The setting up of such a system is application dependent, as it depends on the currently running application what kind of TSN schedule is required for sending messages. As this cannot be described in detail (as it is completely application specific), it is advised for the end user to get into contact with the technology provider (i.e., TTTech Industrial) for support on setting up the overall system.

User Manual

How to use the IWSN system

Once the firmware is uploaded to each IWSN node, these devices will connect automatically to the IWSN network, searching for beacons to detect the IWSN Gateway and providing quality reports to the SDN controller.

Regarding the IWSN Gateway, once the `start.sh` bin bash script has been executed, all the system subcomponents should be available and interconnected. To configure the IWSN Gateway, before the definition of the frontend service, the following API/Rest examples must be used. The complete API/Rest specification will be defined in the final deliverable.

URL: `/api/v1/login`
Method: `POST`
Auth required : NO
Permissions required: None
Data constraints:

```
{
  "username":"admin",
  "password":"admin"
}
```

Success Responses:
Condition: User login success.
Code: *200 OK*

Content:

```
{
  "access_token":"token1",
  "refresh_token":"token2",
  "expires_in":300,
  "token_type":"Bearer"
}
```

URL: `/api/v1/gatewaystatus`
Method: `GET`
Auth required: YES
Permissions required: None
Data constraints: `{}`
Success Responses:
Condition: IWSN Gateway information status.
Code: *200 OK*
Content:

```
{
  "ethernet": {
    "dhcp": "Enabled",
    "ip": "192.168.2.1",
    "state": "connected",
    "mDNS": "i4q-iwsn-16-38.local"
  },
  "wifi": {
    "ip": "192.168.1.2",
    "state": "disabled",
    "ssid": "i4q-iwsn-16-38-wifi"
  },
  "sdn-controller": {
    "state": "connected"
  },
  "lowpan": {
    "ip": "fd00::1",
    "state": "connected",
    "conf_nodes": 5,
    "unconf_nodes": 5,
    "sensors": 8
  }
}
```


URL: `/api/v1/wsn/upload`
Method: `PUT`
Auth required: YES
Permissions required: None
Data constraints: `{}`
Success Responses:

Condition: Upload SDN controller scheduling and sensor configuration to IWSN nodes.
Code: *200 OK*
Content: `{}`

URL: `/api/v1/config/kafka`
Method: `PUT`
Auth required: YES
Permissions required: None
Data constraints:

```
{
  "state":True,
  "broker":"192.168.1.1",
  "port":9092
}
```

Success Responses:
Condition: Kafka subscription success.
Code: *200 OK*
Content: `{}`

How to use the TSN system

As mentioned before, a TSN system cannot be used out of the box and much application specific information is required for setting up and using the overall system. A tool is available for configurating the overall network, enabling the user to define the following aspects of a TSN Network (which is application dependent):

- Network topology
- Planning of new components and data streams
- Network configuration

Similar to the setting up of the TSN system, here much application and technology background is required. Therefore, setting up, configuring and deploying a TSN system can only be performed with the support of experts. For more information, please contact TTTech Industrial.