D3.12 – i4Q Trusted Networks with Wireless and Wired Industrial Interfaces v2

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	D3.12 – i4Q Trusted Networks with Wireless and Wired Industrial Interfaces v2		
WORK PACKAGE	WP3 – BUILD: Manufacturing Data Quality		
DATE OF DELIVERY	CONTRACTUAL 31-Dec-2022	ACTUAL	30-Dec-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 a preceding deliverable called D3.4 “i4Q Trusted Networks with Wireless and Wired Industrial Interfaces” delivered at M18.		
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 final state of solution is performed, showing the relationship between the technical requirements, described in previous deliverables, and the functionalities of the solution.		

Document History

VERSION	ISSUE DATE	STAGE	DESCRIPTION	CONTRIBUTOR
0.1	07-Nov-2022	ToC	Table of Contents and D3.4 previous content available	ITI
0.2	24-Nov-2022	1st Draft	Updated with IWSN contributions.	ITI
0.3	25-Nov-2022	1st Draft	Updated with TSN contributions.	TIAG
0.4	02-Dec-2022	Review	Internal review.	BIBA, RIAS.
0.5	13-Dec-2022	Final Draft	Reviewed version.	ITI, TIAG
1.0	30-Dec-2022	Final Document	Final quality check and issue of final document	CERTH

Disclaimer

Any dissemination of results reflects only the author's view and the European Commission is not responsible for any use that may be made of the information it contains.

Copyright message

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

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 v2** (i4Q^{TN}) Solution providing the general description, the technical specifications and the implementation status. It is an updated deliverable with respect to its previous version *D3.4 Trusted Networks with Wireless and Wired Industrial Interfaces* (i4Q^{TN}). The deliverable **D3.12** 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

This document i4Q **Trusted Networks with Wireless and Wired Industrial Interfaces v2** is an update of v1 of D3.4, for this reason it contains information of the 1st version together with the updates developed in this 2nd version.

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: 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.

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 networks (WSN) with state-of-the-art mechanisms to ensure robustness while operating in industrial environments characterized by a high level of noise and interferences, generated by other technologies at the same frequency band or by reflections caused because of machines or metal surfaces, typical in industrial scenarios. 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, such as IEEE 802.15.4e standard compliant Industrial Wireless Sensor Network (IWSN) nodes for the wireless part of the solution or switches and devices that supports the different specifications of the TSN standard for the wired part. 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, such as payload and beacon encryption, to prevent non-registered nodes to connect to the IWSN.

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 99% of MAC 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.

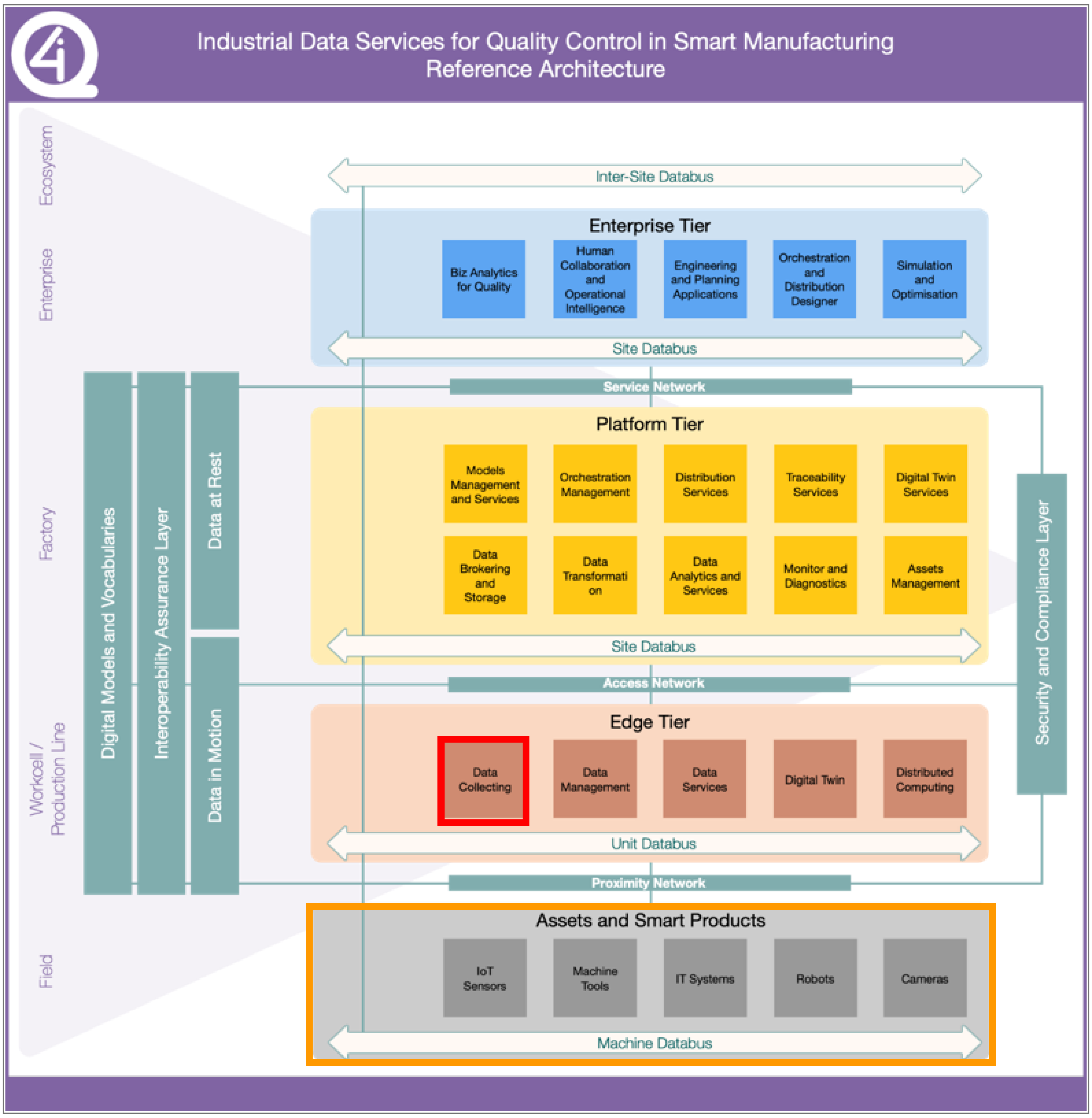

Figure 1. i4QTN Solution Architecture

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 with 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. The controller supports multiple unbound interfaces, such as UART link layer interface with a ISN node or TCP/UDP links over ethernet/wireless for other types of networks.
- **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. **Figure 2** shows the internal architecture of the IWSN developed gateway.

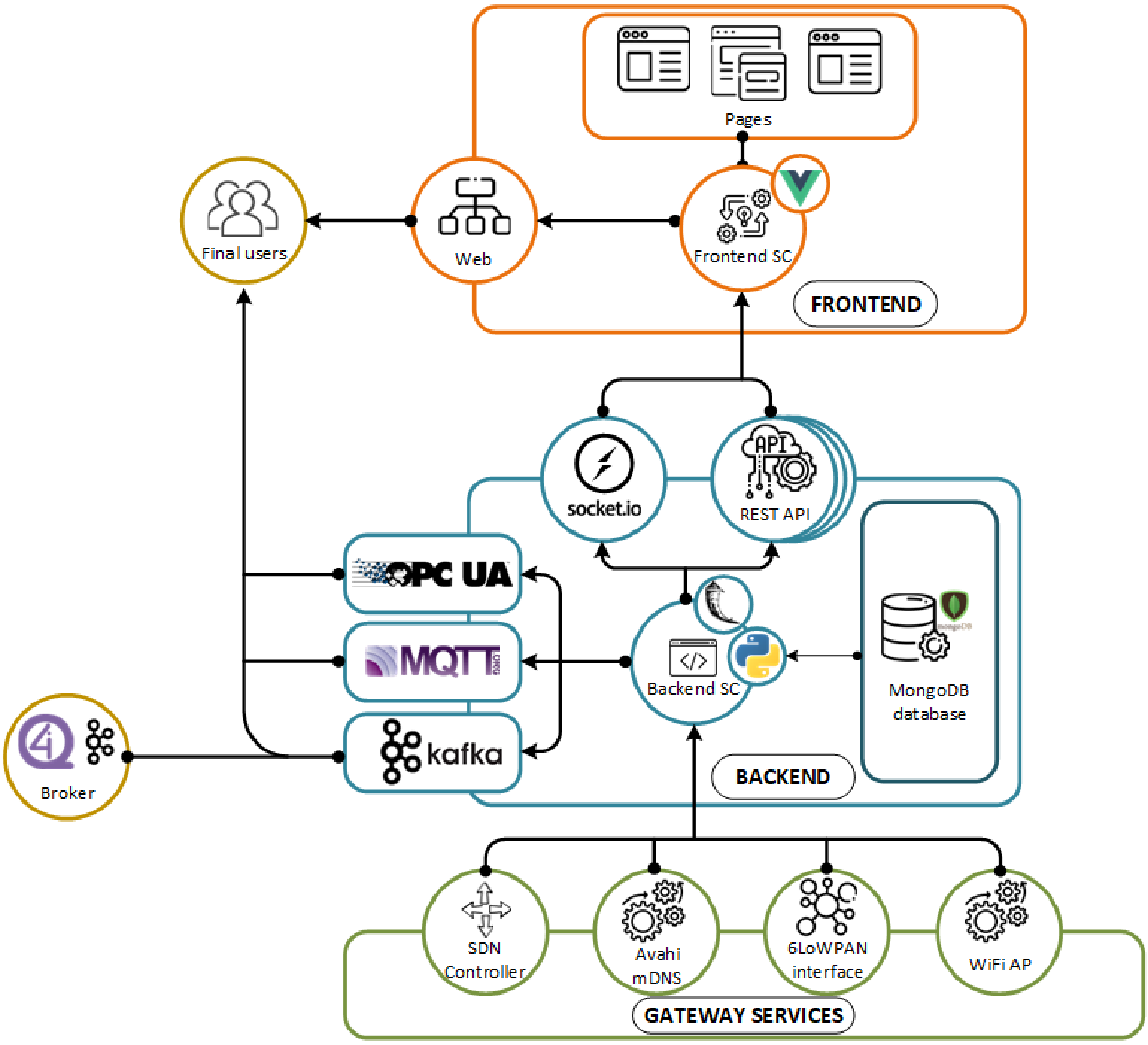

Figure 2. Final IWSN Gateway internal architecture with frontend framework included

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 allows 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 allows 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. Detailed descriptions and screenshots of the web user interfaces are described in the following paragraphs.

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.

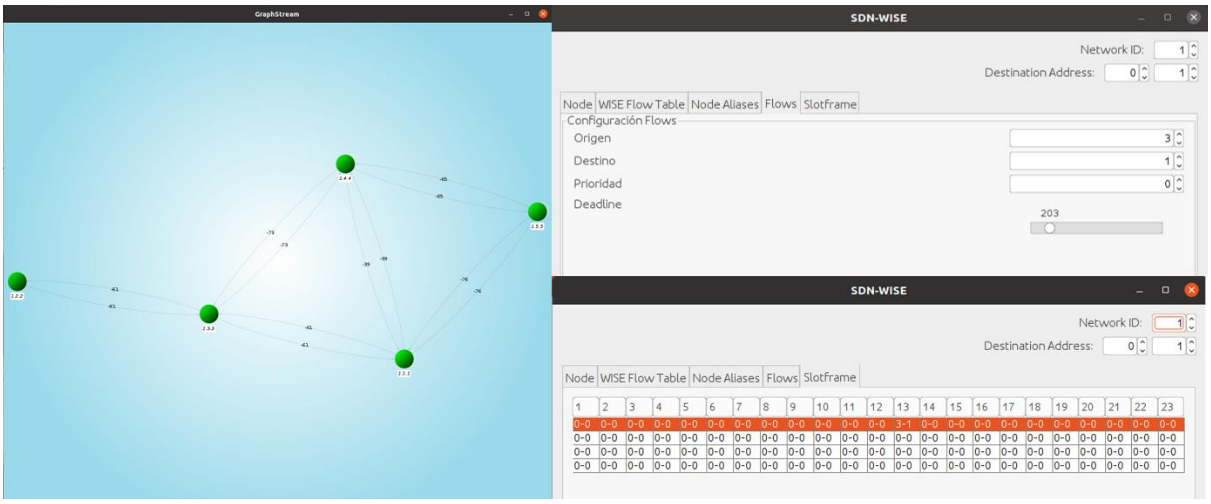

Figure 3. SDN controller and IWSN validation tests

Once all the subcomponents of the system, showed in **Figure 2**, have been deployed, following the procedures explained on annexes, the complete system can be configured and controlled using the REST API interface. To simplify the use of the system, the frontend webserver interface guides the user during different steps to configure the common infrastructure, connects the wireless industrial nodes and its sensors and interconnect with other systems, such as i4Q Kafka broker or even visualizes a streaming of generated data with the webserver itself. Below some screenshots of the configuration process are listed.

First, **Figure 4** shows the home page with common information about the different subcomponents of the IWSN, such as the gateway and SDN controller, information related to each of the wireless nodes and one tool to export all the generated data to a csv file.

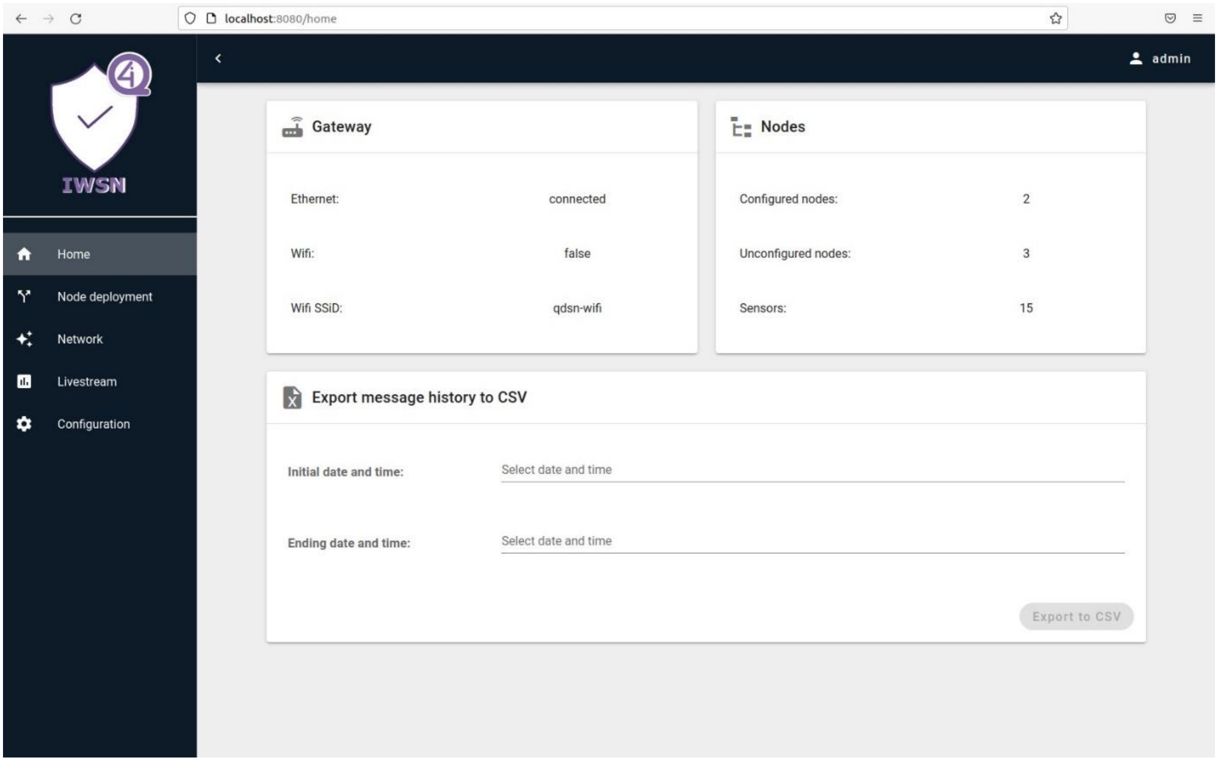

Figure 4. Home page of the IWSN web configuration tool

Second, **Figure 5** shows the node deployment page. This page shows a list of all the wireless nodes connected to the gateway and it is possible to run a wizard that guides the user during 3 steps to correctly deploy the wireless node with enough quality of service, connect the industrial sensor to the platform and configure the parameters acquisition for each sensor.

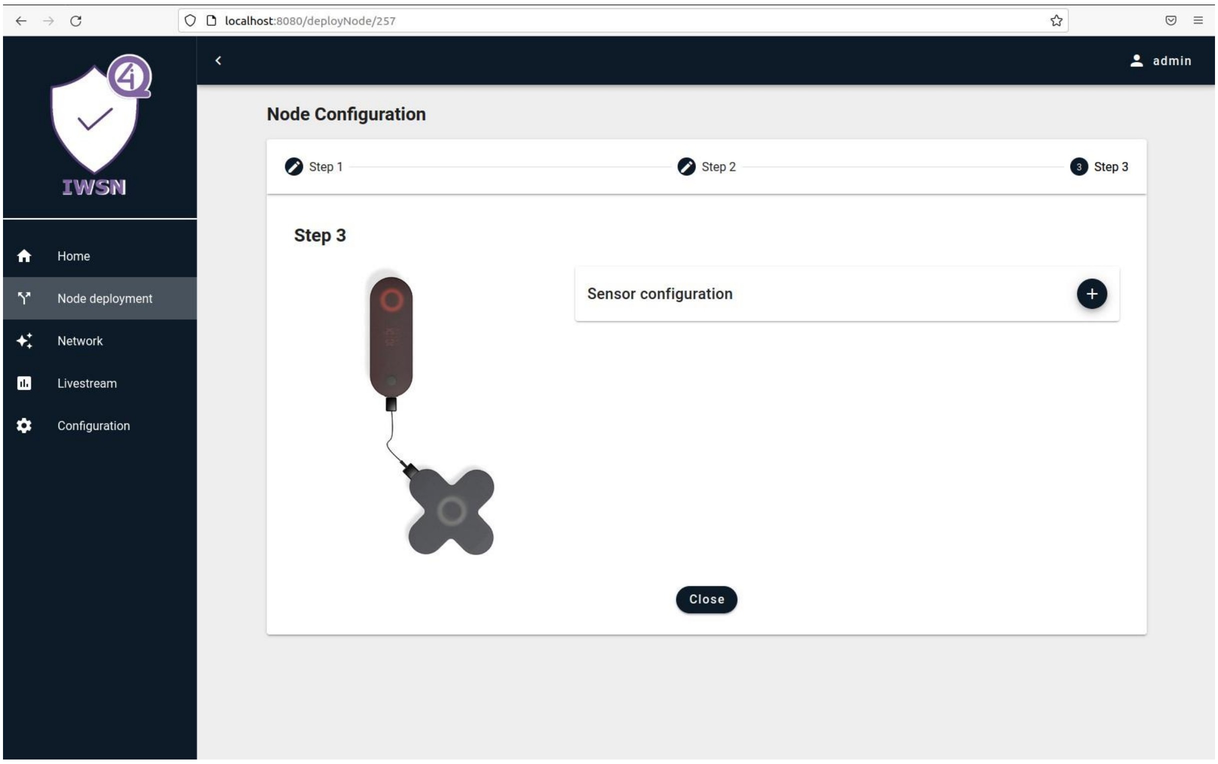

Figure 5. Node deployment page of the IWSN web configuration tool

Third, the platform allows to show network statistic and real time data visualization using the Network and Livestream pages. **Figure 6** shows statistic of the IWSN system and the physical topology of the wireless mesh network.

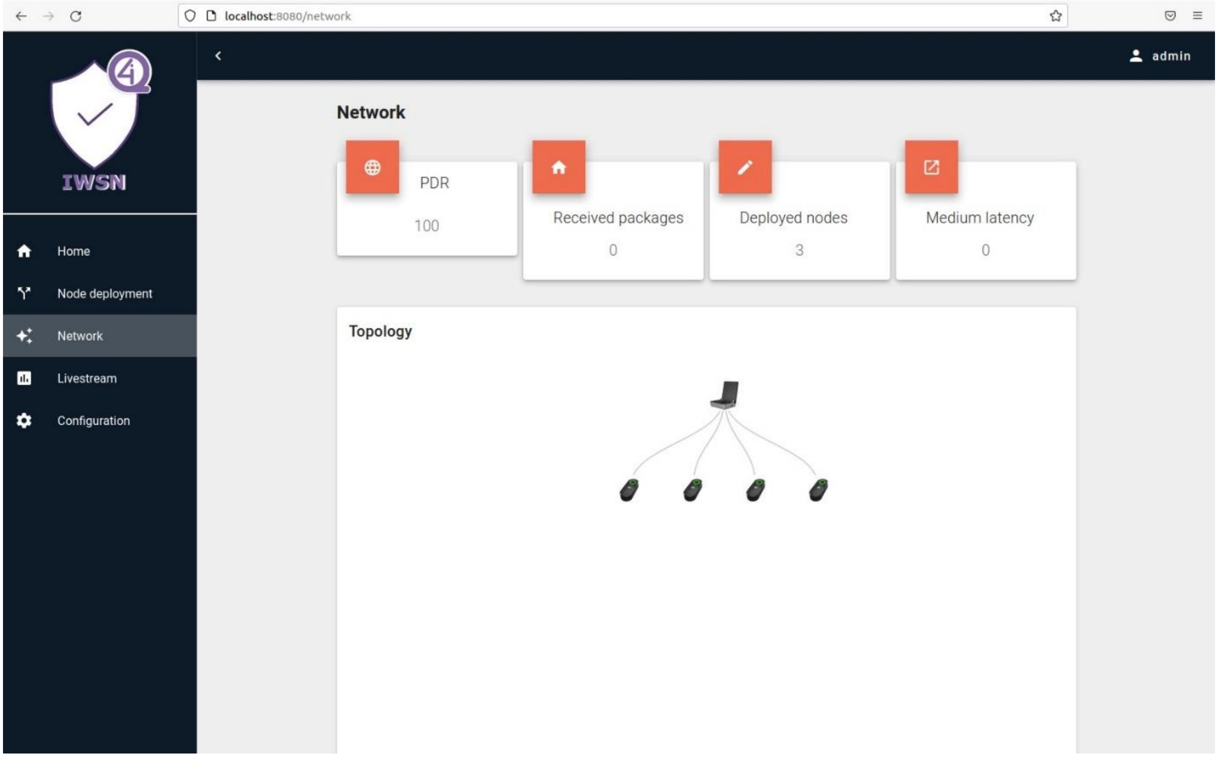

Figure 6. Network page of the IWSN web configuration tool

Finally, there is a configuration page for common system configuration, such as local WiFi access point, network configuration, local database, and integration services with Industry 4.0 protocols like MQTT, OPC-UA and Kafka. **Figure 7** shows the configuration of the Kafka broker locally deployed to simulate the common i4Q broker.

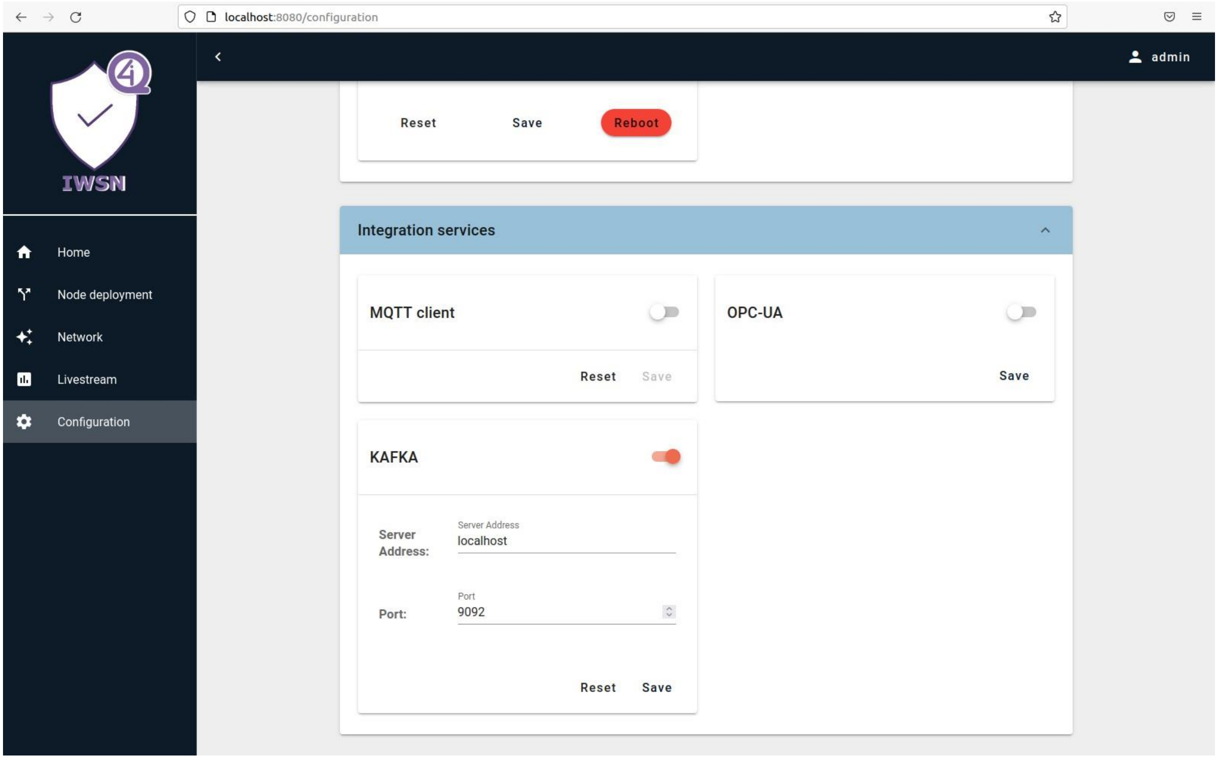

Figure 7. Configuration page of the IWSN web configuration tool

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.

Hardware architecture is based on Texas Instrument CC2538, a 32-bit Arm Cortex wireless microcontroller which is a well-known system for WSN developments and is standard-compliant with the last version of IEEE 802.15.4, supporting the implemented MAC access method TSCH, essential for deployment in industrial scenarios. The microcontroller is powered by different industrial sensor interfaces, optocouplers to prevent high voltages damages and different voltage regulators, supporting different power supply from 3V3 to 24 V.

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 8. Left: HUB platform v0.0.1; Middle: HUB and node combined in v0.1.0; Right: Third iteration solving some issues of the previous version

Figure 8 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 middle, 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. On the right, the third iteration of the hardware development, solving some issues of the previous version related to the RS-485 translation to UART communication and the ADC of the 4-20mA, that was not able to support 24V. The third iteration solve all the identified problems of the previous version, but the SPI interface to the e-Paper display has not been validated before M24. The information planned to be represented with this display now is available via the webserver user interface.

Figure 9 shows the tags that identifies the different interfaces integrated in each of the wireless nodes, including 2 digital inputs of 24 V opto-coupled, 2 analog 2-40 mA inputs, 2 analog 0-10 V inputs, four pins connector for I2C sensors and another four pins connector for RS-485 industrial bus, allowing to connect with devices like Modbus-RTU analyser.

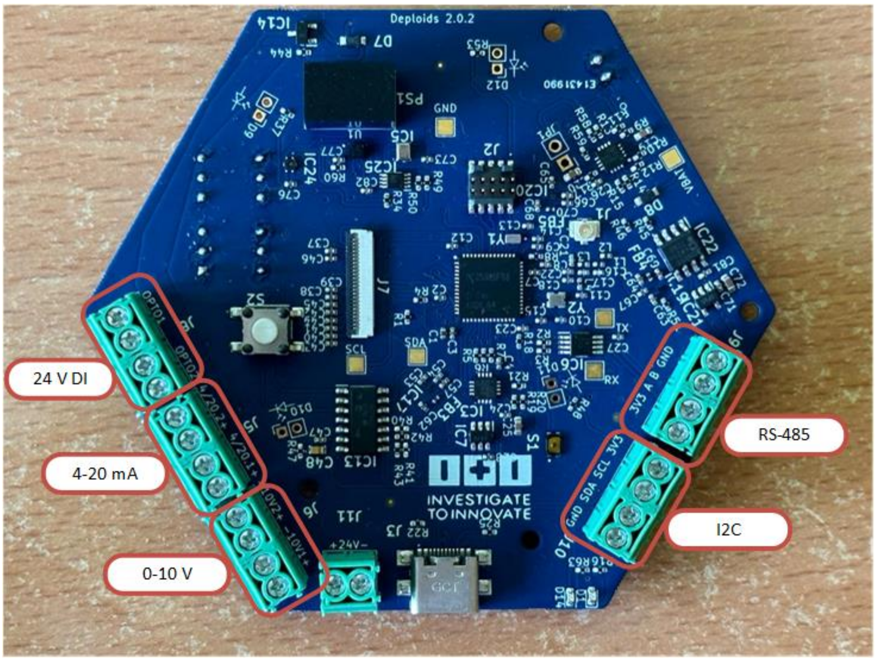

Figure 9. Detail of different industrial interface of the Hardware platform

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.

Compared to the solution presented in deliverable D3.4, the aim is to provide a network-wide scheduling and re-planning tool that is capable of identifying the needed quality of service configuration settings to achieve application specific latency and jitter requirements for the network layer; this capability needed for the scheduling / configuration of the network to allow to automatically implement a large fault-tolerant industrial system using the following steps:

- Specify industrial network requirements and generate the network scheduling architecture link constraints of the industrial network configuration (determinism boundaries, bandwidth restrictions, etc.),
- Apply scheduling algorithms (also called network planning algorithms) to ensure availability of bandwidth for critical traffic and determinism,
- Allow dynamic reconfiguration based on quality of service (QoS).

This configuration software automatically sets up all standard network components (network switches and end systems) via a defined communication protocol (e.g., NETCONF) allowing to hide network planning complexity from the user and enabling execution of reconfiguration changes in an automated manner, e.g., dynamic addition / removal of publishers and/or subscribers in a system. The software does not require any user interface, it simply runs in a network device and facilitates the planning of data streams according to the publisher / subscriber requirements. The configuration itself is achieved using a network planner as described. Planning based on a network topology and capability database can be supplied via offline design or created by online discovery of the network topology and capabilities of the devices. As soon as a stream publisher and at least one subscriber are online, the configuration for all Deterministic Ethernet-based devices, e.g., TSN (Time-Sensitive Networking), on the networking path of that specific stream is generated and deployed to the network nodes. In case of dynamic changes to the TSN network (Hot-removal and/or addition of devices participating in stream-based communication), an incremental planning calculation allows to "upgrade" the schedule without interfering with already established communication connecting within the network. This way communication does not need to be interrupted or paused for network expansion, or reduction. This approach deploys Centralized Network Configuration (CNC) and Centralized User Configuration (CUC) servers to control a set of Ethernet-based end systems and switched endpoints within a TSN network. CNC/CUC planning features can support a user e.g., in the following constraints: (i) guaranteeing end2end latency, (ii) ensuring a certain transmit time window constraint for time-critical or phase-aligned calculations of a distributed control application, (iii) adding incrementally new data streams and new devices without affecting existing scheduled streams, etc. The standard that deals with the (re-)configuration of TSN networks is IEEE 802.1Qcc. This standard, already introduced in the previous deliverable D3.4, is an enhancement of the Stream Reservation Protocol (SRP) (IEEE 802.1Qat) designed for the resource management in networks using the Credit Base Shaper (CBS) (IEEE 802.1Qav). The supporting IEEE standards considered in the planner are:

- IEEE 802.1Qbv Time Aware Shaping: The core of TSN is a time-triggered communication principle, known in TSN as the "Time-Aware Shaper" (TAS), which deterministically schedules traffic in queues through switched networks. This is standardized in IEEE 802.1Qbv. The TAS enables to control the flow of queued traffic from a TSN-based bridge. 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 egress ports during the scheduled time windows. Other queues and ports are then blocked, thereby removing the chance of scheduled traffic being interrupted by non-scheduled traffic.
- IEEE 802.1Qcp Bridges and Bridged Networks Amendment: YANG Data Model: This amendment specifies a Unified Modelling Language (UML) based information model and a YANG data model that allows configuration and status reporting for bridges and bridge components including MAC bridges, Two-Port MAC Relays (TPMRs), Customer Virtual Local Area Network (VLAN) bridges and Provider Bridges. It additionally defines the relationship between the information and the data model and the models for the management capabilities.
- IEEE 802.1CB Frame Replication and Elimination: The IEEE 802.1CB standard implements a redundancy management mechanism similar to the approaches known from HSR (High-availability Seamless Redundancy – IEC 62439-3 Clause 5) and PRP (Parallel Redundancy Protocol – IEC 62439-3 Clause 4). In order to increase availability, redundant copies of the same message are communicated in parallel over disjoint paths through the network.
- IEEE 802.1Qci Filtering and Policing: Per stream filtering and policing prevents adverse effects on system communication performance, as a result of faulty end-stations or

inconsistently configured software which could violate the engineered bandwidth use. This fine-grained policing capability allows to better control different data flows in complex systems.

The configuration tool (see **Figure 10**) is an embedded software solution that interfaces with OPC UA applications to provide planning and (re-)configuration of TSN networks and devices. The tool doesn't require any user interface anymore, as it simply runs directly on one of the devices within the network and facilitates the planning of data streams according to the requirements of OPC UA publishers and subscribers. The final configuration of the individual network components is achieved using standard YANG models and NETCONF [4].

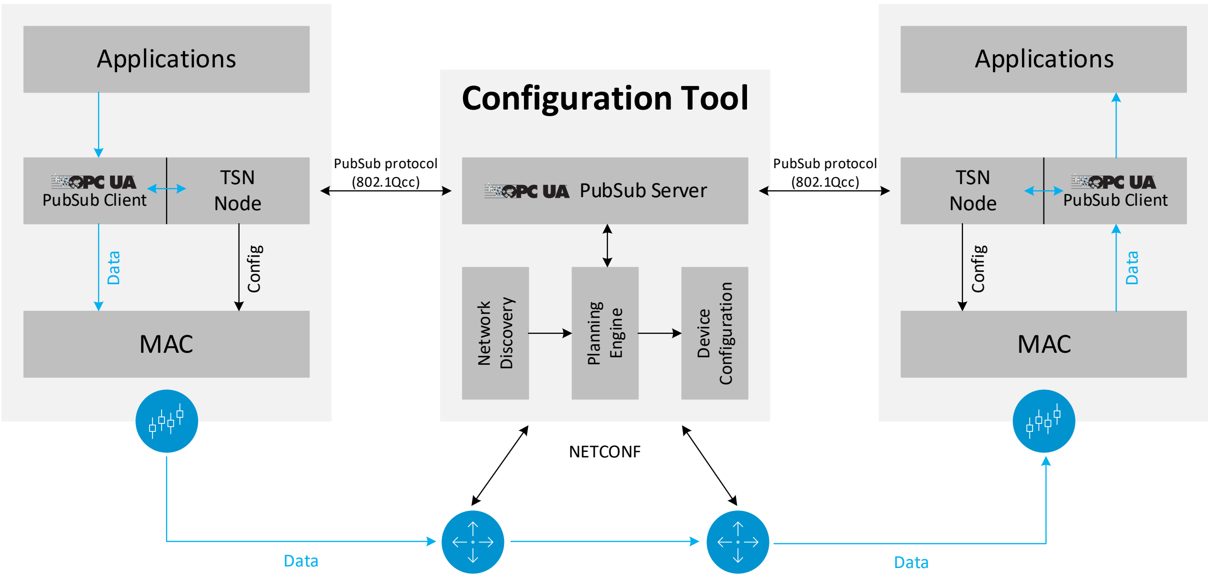

Figure 10: Embedded (re-)configuration tool

One of the main elements for the (re-)configuration of deterministic communication networks is the User Network Interface (UNI). On the user side of this interface are the talkers and the listeners, whereas on the network side of the interface are the bridges [3]. The concept of the UNI is that the user specifies the requirements for the streams that they want to transmit without having all the details about the network.

One of the main elements of the (re-)configuration of TSN networks is the User Network Interface (UNI). On the user side of this interface are the talkers and the listeners, which are in principle the end devices within a TSN network. On the network side of the interface are the bridges, which are the switches in the network [3]. The concept of the UNI is that the user specifies the requirements for the stream that they want to transmit over the network, without having all the details of the current network. The network then analyses this information with network capabilities and configures the bridges to meet the user requirements [5]. IEEE 802.1Qcc (which was already introduced in deliverable D3.4) defines three configuration models with regards to their architecture, which provides the realization of the configuration paradigm [3]:

- **Fully Distributed Model:** In this model, the User Network Interface (UNI) is situated between the talker/listener and the bridge to which it is connected (network). The user transmits its requirements, and the network propagates them through the relevant paths.

The management of the bridges is performed locally just with the information that is available to that bridge. One limitation of this model is the lack of a centralized view with complete knowledge of the network, that makes it not suitable for the configuration of the Time Aware Shaper (IEEE 802.1Qbv).

- **Centralized Network / Distributed User Model:** This model tackles the limitation of the fully distributed model by introducing the Centralized Network Configuration (CNC). The UNI is still between the talkers/listeners and the bridges, but in this model, the bridge communicates the user requirements directly to the CNC. The CNC has a complete knowledge of the network topology as well as the bridges capabilities and that enables it to perform more complex calculations needed to configure the Time Aware Shaper, frame replication and elimination or frame pre-emption. The management of the bridges is performed by the CNC using a network management protocol. The management of end devices is not performed by the CNC. The CNC can either exist in an end device or in a bridge.
- **Fully Centralized Model:** The configuration of the end devices was not addressed in the two previous models. There are, however, use-cases for highly critical applications (automotive or industrial), where complex timing requirements are needed that need extra configuration. Especially for those cases, the notion of the Centralized User Configuration (CUC) is introduced. The talkers/listeners communicate their requirements to the CUC and then the CUC exchanges this information with the CNC using the UNI. The CNC and the CUC can be implemented in the same device or in separate devices. A graphical depiction of the IEEE 802.1Qcc Fully Centralized Model is provided in **Figure 11**.

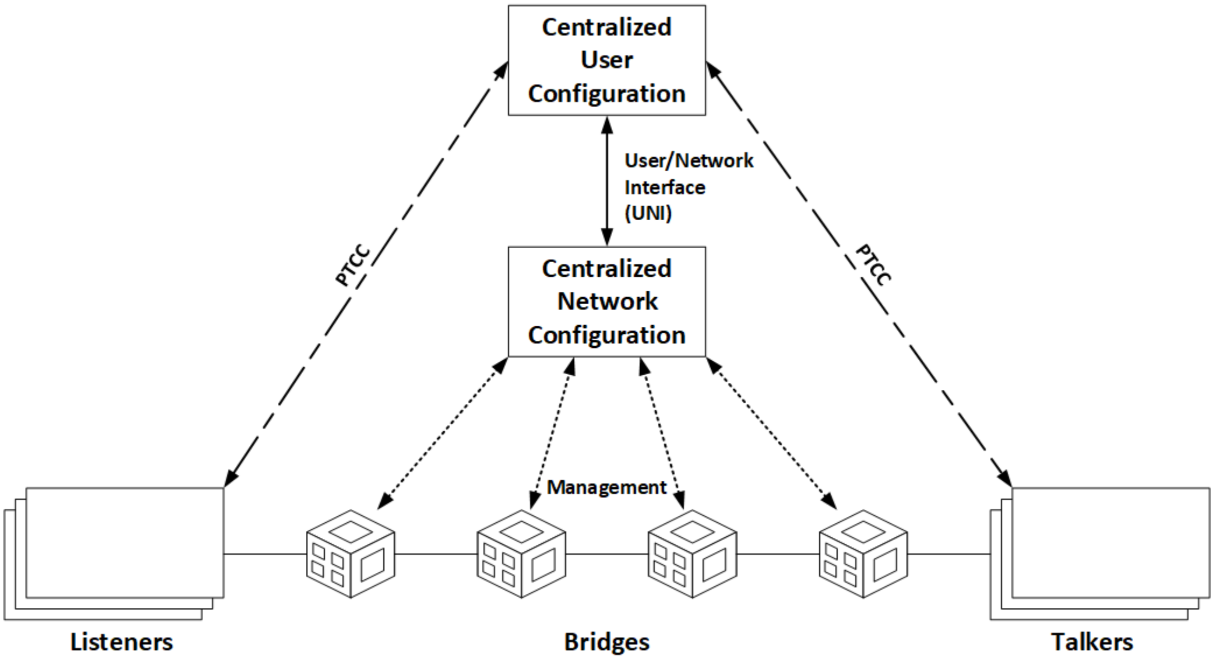

Figure 11: IEEE 802.1Qcc Fully Centralized Model

The (re-)configuration tool includes a CUC server interface for automated stream requests from OPC UA stacks, which is handled by the PTCC. PTCC stands for PubSub TSN Centralized Configuration and is a proposal for a CUC using OPC UA as interface to the users. It defines the configuration interface between the CUC and the OPC UA PubSub end station (listeners and talkers). The PTCC is part of the CUC implementation for PubSub TSN based systems using the TSN fully centralized configuration model, described before. The PTCC configuration interface is based on a client-server type of operation, having the server placed in the CUC, which can be in one of the bridges, and the clients in the different endpoints.

The workflow of the (re-)configuration of the TSN network is the following approach, starting with the initial configuration of the actual network. Hereby it is assumed that the CNC doesn't have any knowledge about the overall network. Additionally, the end systems (which are the talkers and the listeners) don't actively notify the CNC about their communication needs. The CNC has to capability to acquire the necessary knowledge from the network. The initial configuration is as follows:

1. The CNC establishes a connection with the available bridges in the network using the NETCONF protocol. By using this connection, the CNC is aware of the capabilities of the different bridges.
2. The CNC uses the Monitor and Extractor part to extract information from the bridges that are capable of learning information from the network. Exploiting the NETCONF protocol, the CNC changes the behaviour of the bridges, so they are capable of identifying the changes within the network.
3. The selected bridges send the detected and updated information to the CNC.
4. The CNC collects all the information, and as soon as enough information has been collected, it creates an initial configuration for the total network, including a schedule for the (deterministic) communication.
5. The CNC distributes the new configuration of the bridges and the CUC. It uses the NETCONF protocol to communicate the new configuration to the different bridges and the UNI interface to communicate it to the CUC.

After the initial configuration has been distributed over the total network, the bridges are now operating in a mode that enables them to identify or learn about changes in the network. Possible changes could be either changes in the topology of the network or changes in the messages to be sent over the network. If a modification occurs in the network, e.g., changes in the traffic caused by the addition or removal of devices, the CNC has two possibilities of identifying these modifications. The first option is by communicating with the CUC using OPC UA (exploiting the PTCC). The second option is through the “learning” capabilities of the bridges that are in contact with the CNC using the LLDP protocol. This could also be done periodically in fixed periods. If a modification or change has been detected in the network, the abovementioned steps 4) and 5) of the initial configuration process will be repeated [5].

The solution that is being provided to establish dynamic (re-)configuration under full consideration of deterministic requirements and changing environment with real-time constraints will provide intelligence to the deterministic network. This is established by creating a semi-autonomous system that is capable of learning the characteristics of the network by continuous monitoring and analysing the network. The system will be able to update the configuration of the network, when e.g., end systems are being added or removed from the network, by preserving the original real-time requirements (e.g., maximum transmission latency and jitter) and potentially improve them (e.g., less latency and jitter).

The system is made up of the following functional elements [3]:

- **Monitor**: This part observes the network's traffic and collects various measurements to identify traffic patterns. The overall aim is to recreate from the identified parameters the original real-time constraints defined by the currently running applications.
- **Extractor**: This element derives the traffic parameters based on the traffic patterns observed by the Monitor and the previous knowledge of the network and applications that is already available. This part is the learning phase of the (re-)configuration.
- **Scheduler**: This part uses the traffic parameters (acquired by the Extractor) to generate a new schedule for the overall network, that maintains and/or improves the deterministic guarantees.
- **Reconfigurator**: Finally, this element is in charge of updating the overall network configuration, so that the network follows the new communication schedule, which was generated by the Scheduler.

The whole system is depicted in **Figure 12**.

The fully centralized network model, introduced before and proposed in the standard IEEE 802.1Qcc, fits very well with the view that the configuration system calculates the new configuration for the network. The Extractor and the Scheduler components are implemented as part of the CNC, thereby being capable of having an overall view of the whole network. The CNC needs to have some scheduling capacity if it is meant to configure the Time Aware Shaper (TAS).

Additional to the generic definition of the UNI, TSN also provides a concrete realization of the interface for the centralized models. This formalization is defined by using the data modelling language YANG. These YANG models can be communicated between the Talkers and Listeners or the CUC and the CNC by protocols like e.g., NETCONF or RestConf [6].

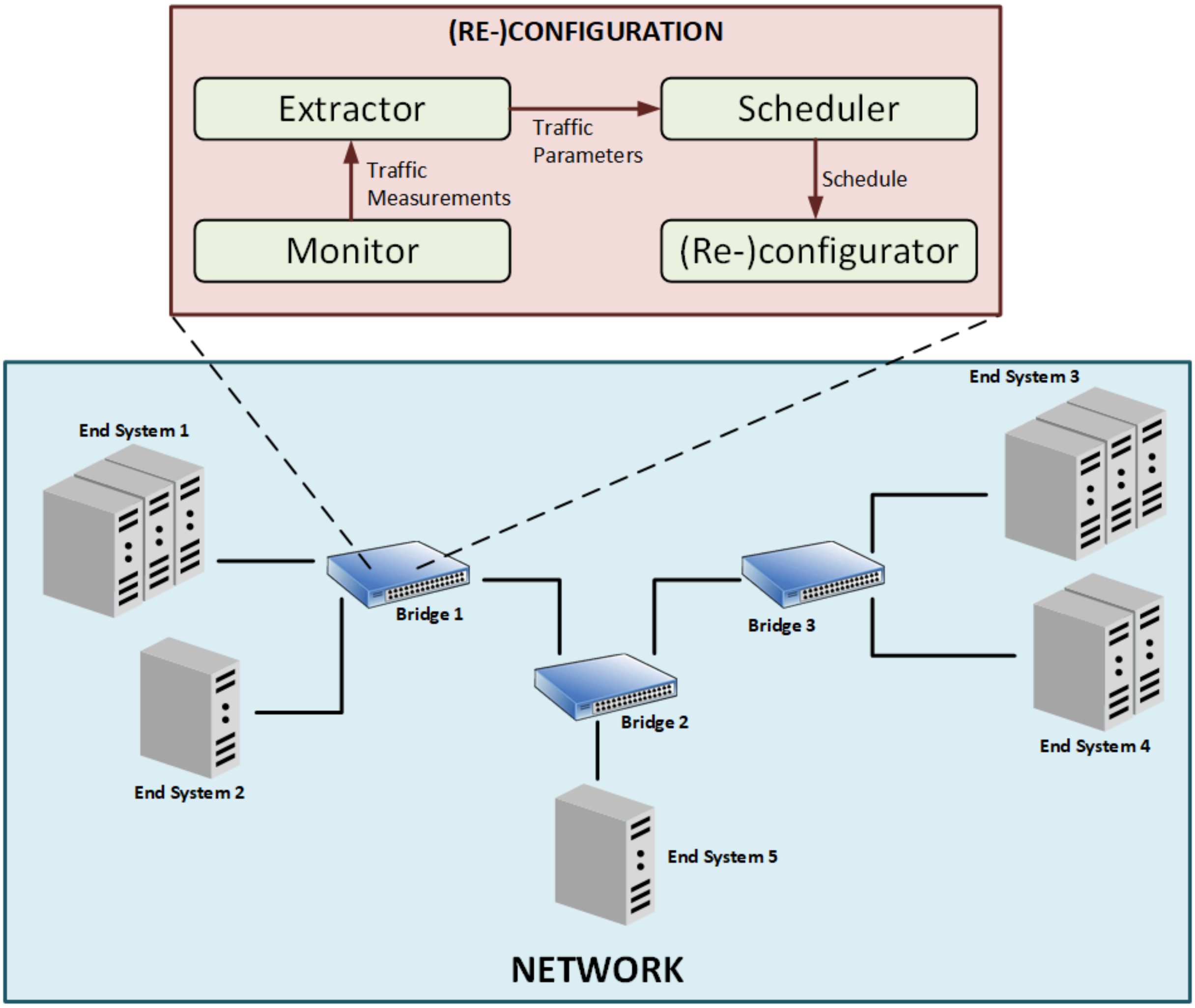

Figure 12. (Re-)configuration System – Identifying changes in the network

As a temporary demonstrator for this technology, we have created in our lab a setup of a TSN network containing multiple end systems and bridges (see **Figure 13**). With this test setup, we want to demonstrate the automatic (re-)configuration of a TSN network using OPC UA. The network is setup in such a way, that end systems can be dynamically added or removed from the network by activating or de-activating them. The network (re-)configuration management system is hosted on the PC connected to the network.

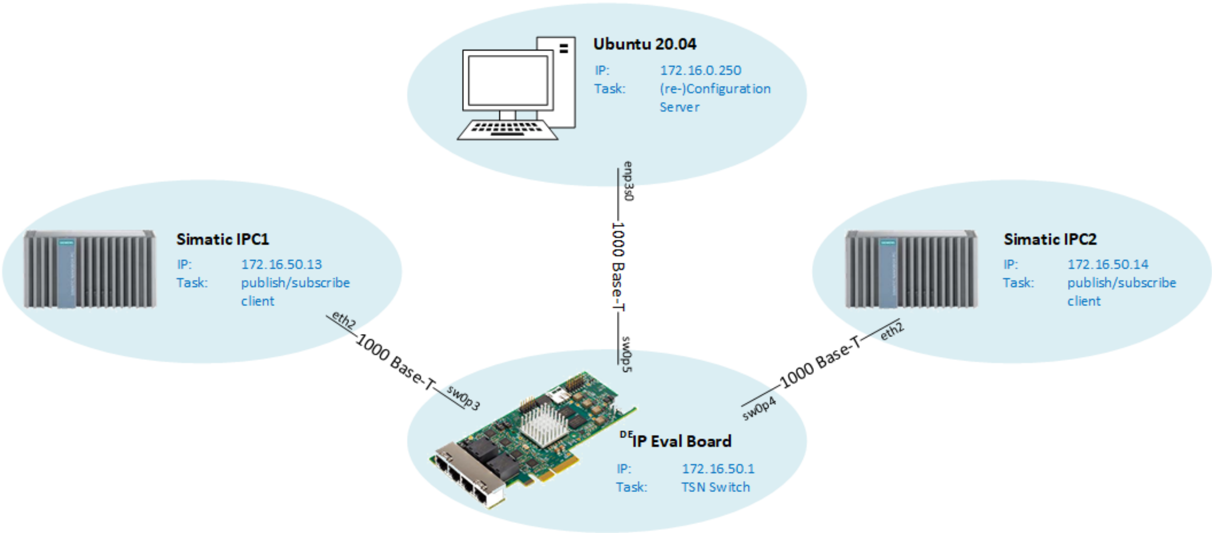

Figure 13. (Re-)configuration System – Setup

3.1.1 Solution features analysed and mapping with user requirements

A set of features has already been developed for i4QTN, based on the set of user and pilot requirements referring to i4QTN [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 i4QTN 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 i4QTN.

- *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 schedule 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 History

This section provides the version history of the i4Q[TN] solution implementation up to M24, 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.
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.
V0.1.1 (HW)	10/10/2022	Fixed some issues with RS485 translator and ADC chipset.
V0.1.1 (SW)	10/10/2022	Kafka producer integration
V0.1.2 (SW)	18/11/2022	Web user interface development
V1.0.0 (HW)	23/12/2022	Third and final version of the Hardware platform updated for M24 solution release.
V1.0.0 (SW)	23/12/2022	M24 solution release

Table 1. i4Q[TN] version history

4. Conclusions

Deliverable "D3.12 - i4Q Trusted Networks with Wireless and Wired Industrial Interfaces v2" 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 the complete development period of the solution are described in detail in this document, showing the final status of the i4Q^{TN} solution up to M24.

References

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

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

[3] AUTOWARE (2018), D2.2a AUTOWARE Deterministic Ethernet Communications. Available: https://www.autoware-eu.org/deliverables/D2.2a,pdf

[4] TTTech Industrial (2022), Introducing Slate YNS. Available: https://www.tttech-industrial.com/products/slate/slate-yns/

[5] Gutiérrez, M., Adamaj, A., Steiner, W., Dobrin, R., & Punnekkat, S. (2017, September), Self-Configuration of IEEE 802.1 TSN networks. In 2017, 22nd IEEE international conference on emerging technologies and factory automation (ETFA), pp. 1-8), IEEE

[6] Bello, L. L., & Steiner, W. (2019). A perspective on IEEE time-sensitive networking for industrial communication and automation systems. Proceedings of the IEEE, 107(6), 1094-1120.

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