D6.10 – i4Q Solutions Demonstrator v3

WP6 – EVALUATE: Piloting and Demonstrating

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	41 months
PROJECT URL	https://www.i4q-project.eu/		
DELIVERABLE	D6.10 - i4Q Solutions Demonstrator v3		
WORK PACKAGE	WP6 – EVALUATE: Piloting and Demonstrating		
DATE OF DELIVERY	CONTRACTUAL 31-Jan-2024	ACTUAL	31-Jan-2024
NATURE	Demonstrator	DISSEMINATION LEVEL	Public
LEAD BENEFICIARY	EXOS		
RESPONSIBLE AUTHOR	Arcadio Garcia		
CONTRIBUTIONS FROM	1-CERTH, 2-ENG, 3-IBM, 4-ITI, 5-KBZ, 6-EXOS, 7-IKER, 8-BIBA, 9-UPV, 10-TUB, 11-UNI, 12-TIAG, 13-CESI, 14-AIMP		
TARGET AUDIENCE	1) i4Q Project partners 2) industrial community 3) other H2020 funded projects 4) scientific community		
DELIVERABLE CONTEXT/ DEPENDENCIES	This document has relationship to other preceding documents as follows:		

	• D3.1 - i4Q Data Quality Guidelines • D3.2 - i4Q QualiExplore for Data Quality Factor Knowledge • D3.3 - i4Q Blockchain Traceability of Data • D3.4 - i4Q Trusted Networks with Wireless & Wired Industrial Interfaces • D3.5 - i4Q Cybersecurity Guidelines • D3.6 - i4Q IIoT Security Handler • D3.7 - i4Q Guidelines for Building Data Repositories Industry 4.0 • D3.8 - i4Q Data Repository • D4.1 - i4Q Data Integration and Transformation Services • D4.2 - i4Q Services for Data Analytics • D4.3 - i4Q Big Data Analytics Suite • D4.4 - i4Q Analytics Dashboard • D4.5 - i4Q AI Models Distribution to the Edge • D4.6 - i4Q Edge Workloads Placement and Deployment • D4.7 - i4Q Infrastructure Monitoring • D4.8 - i4Q Digital Twin • D5.1 - i4Q Data-Driven Continuous Process Qualification • D5.2 - i4Q Rapid Quality Diagnosis • D5.3 - i4Q Prescriptive Analysis Tools • D5.4 - i4Q Manufacturing Line Reconfiguration Guidelines • D5.5 - i4Q Manufacturing Line Reconfiguration Toolkit • D5.6 - i4Q Manufacturing Line Data Certification Procedure **D6.10** provides inputs for **D6.17** and **D6.18**
EXTERNAL ANNEXES/ SUPPORTING DOCUMENTS	None
READING NOTES	None
ABSTRACT	This deliverable **D6.10 v3**, on the one hand, extends and completes the technical information of the Module Stations described in the deliverable D6.7, providing detailed information, such as a Generic Pipeline, server configuration in terms of all the necessary software infrastructure, network characteristics, connection parameters, etc. All this information is necessary for the i4Q Solutions Providers to test the i4Q Solutions under the specific conditions of the experiment. The second part of D6.10 v3 is dedicated to collect the results of the Test Cases executed by each i4Q Solution, for which a standard procedure is used in which, by means of a table, the previous conditions of the Test, its execution results and conclusions are collected.

Document History

VERSION	ISSUE DATE	STAGE	DESCRIPTION	CONTRIBUTOR
0.1	18-Jul-2023	ToC	First ToC	EXOS
0.2	04-Sep-2023	1st Draft	First working document	EXOS, UPV
0.3	22-Sep-2023	2nd Draft	Test Cases Methodology	EXOS
0.4	14-Nov-2023	3rd Draft	Update of technical information of the module Stations by UPV and Test Cases by i4Q Solution Providers	UPV, All i4Q Solution Providers
0.5	21-Dec-2023	4th Draft	Update of Test Cases by i4Q Solution Providers	All i4Q Solution Providers
0.6	02-Jan-2024	5th Draft	Revision all i4Q Solutions Test Cases	EXOS
0.7	10-Jan-2024	6th Draft	Revision all i4Q Solutions Test Cases	All i4Q Solution Providers
0.8	14-Jan-2024	7th Draft	Revision all i4Q Solutions Test Cases	EXOS, UPV, All i4Q Solution Providers
0.9	19-Jan-2024	Internal Review	Internal Review Process	UNINOVA, FIDIA
0.10	22-Jan-2024	Final Draft	Address comments from the internal review	EXOS
1.0	31-Jan-2024	Final Doc	Quality check and issue of final document	CERTH

Disclaimer

Copyright message

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS/ACRONYMS

AD	Analytics Dashboard
AI	Artificial Intelligence
API	Application Programming Interface
BC	Blockchain
CP-AM-CAM	Camera inspection
CP-AM-iDRILL	iDrilling
CP-AM-MAGBACK-BLACK	Magazine
CP-AM-MPRESS	Muscle press
CP-AM-OUT	Output
CP-F-ASRS32-P	CP Factory High-bay storage for pallets
CP-F-FBRANCH	CP Factory Branch
CP-L-BRANCH	CP Lab Branch
CPK	Process Capability
CPU	Central Processing Unit
DA	Data Analytics
DB	Database
DIT	Data Integration and Transformation
DR	Data Repository
DT	Digital Twin
FMU	Festo Module Unit
gRPC	gRPC Remote Procedure Calls
ID	Identification
IIoT	Industrial Internet of Things
IM	Infrastructure Monitoring
IWSN	Industrial Wireless Sensor Network
LRT	Line Reconfiguration Toolkit
MES	Manufacturing Execution System
ML	Machine Learning
MQTT	Message Queuing Telemetry Transport
N/A	Not Applicable

OPC	Open Platform Communications
PA	Prescriptive Analysis
PLC	Programmable Logic Controllers
PQ	Process Qualification
QC	Quality Control
QD	Quality Diagnostics
RA	Reference Architecture
Robotino	Robotino
SDN	Software Defined Networks
SQL	Structured Query Language
SSL	Secure Sockets Layer
TBD	To Be Defined
TCP	Fieldbus communication
TLS	Transport Layer Security
TN	Trusted Networks
TSN	Time Sensitive Networks
UA	Unified Architecture
UI	User Interface
WP	Work Package
WSN	Wireless Sensors Technologies

Executive summary

In this deliverable D6.10 v3 the Module Stations technical information is extended with the aim of to show i4Q Solutions Providers all the characteristics of the industrial experimentation facilities of the UPV i4Q partner, providing an experimental environment in which specific tests of the i4Q Solutions are conducted, once they are already deployed and working in the i4Q Pilots.

To perform the Test Cases, a standard methodology is used that clearly and very synthetically defines the steps to be followed to perform the experiment or test. Test Cases consist of a series of very specific steps and under the consideration of very specific parameters that i4Q Solutions Providers execute to test their i4Q Solutions in certain scenarios.

Each Test Case includes a series of pre-conditions, test data, results and post-conditions that allow to determine the success or failure of each Test Case. All the steps of a Test Case are aimed at testing the functionality and applicability of each Test, based on the preconditions and expected results. A test case is considered the smallest unit of a test and contributes to the overall test script.

The following issues must be considered as being important:

- Firstly, it is an experimental and testing environment with technical characteristics of simulation with a learning approach, so Module Stations are not a real industrial production environment such as those that exist in the industrial production environment in the i4Q Pilots' factories, and therefore their technical characteristics are limited and i4Q Solutions Providers need to be aware of their scope.
- Secondly, there are certain difficulties in obtaining the information that i4Q Solutions Providers may require in order to perform their tests, but it is also not their objective, i.e. they are not the only ones that i4Q Solutions Providers can use. It must be very aware that the Test Cases are very specific experiments and very limited to certain aspects of the i4Q Solutions.
- And finally, with regard to the schedule of execution of Test Cases and given the special characteristics of accessibility to the test environment, i.e. the server and the module stations, it is necessary to know beforehand the and module stations, it is necessary to know in advance when each i4Q Solutions Provider wants to test its Solution, as it will not be possible to test simultaneously between several Solutions, except in those cases where the i4Q Solution Providers deem it necessary due to the specific characteristics of the Test Case.

Document structure

Section 1: Introduction: Provides an introduction to deliverable D6.10 – i4Q Solutions Demonstrator v2.

Section 2: Experimentation Facilities – General Pipeline: Describes all technical details, characteristics of the Industrial Module Stations that are part of the experimental facilities for i4Q Solutions, as well as other auxiliary hardware and software systems necessary for their operation and experimentation. Individual Module Stations, Physical (sensing), Network (communication), Data from Module Stations.

Section 3: Test Cases - i4Q Solutions Testing and analysis of results: In this section, i4Q Solution Providers identify those Module Stations where their Solutions can be tested and give a brief description of their rationale as well as desirable future possibilities.

Section 4Σφάλμα! Το αρχείο προέλευσης της αναφοράς δεν βρέθηκε.**: Conclusions:** The respective conclusions and remarks are presented in this section.

1. Introduction

In general, to initiate a test case, first the actions and parameters to verify the expected behaviour of a test must be described, whereby sets of conditions and variables are defined to determine the quality and success of the software system, and the final results that can confirm these facts.

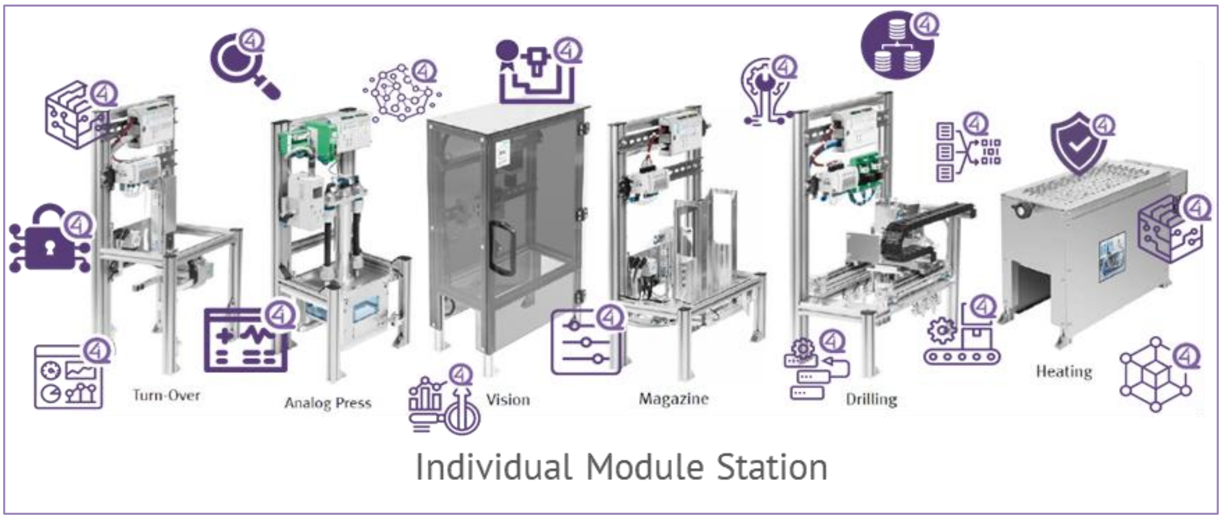

Figure 1. i4Q Solutions Implementation - Schematic representation

The **types** of test cases that will help to understand the purpose of each test case and can be carried out are as follows:

- **Functional**: Used to determine whether the i4Q Solution's functionality meets the Pilots' expectations, in other words, a functional test case is based on the specifications of the i4Q Solution to be tested and identifies whether or not the functionality expected by the i4Q Pilots is successful or not.
- **User Interface**: Verifies aspects related to the graphical user interface, identifying and testing link errors, the look and feel of the i4Q Solution and other aspects that users see or interact with. The main elements affected are the browsers.
- **Performance**: Checks the functionality and response time after executing an action of the i4Q Solutions. One of the outstanding features of the i4Q Solutions is that they work by reading data in very short time intervals and response times must also be very short, so this type of test can be very relevant.
- **Integration**: Determines how the different components of each i4Q Solution interact with each other, and how the i4Q Solutions interact with each other. The aim is to ensure efficient operation between component interfaces, either within an i4Q Solution or between i4Q Solutions.
- **Usability**: Ensures the usability of the i4Q Solutions in a structured and well-defined way, in other words, the series of actions that users must perform in certain specific tasks/steps within each i4Q Solution.

- **Database**: Consists of database testing through a set of actions to test the performance, security and functionality of the database system, verifying that the code written by the i4Q Solution Providers stores and handles data securely, ensuring the functionality of the database system and that it can handle the expected volume of data without errors or data loss.
- **Security**: Identifies weaknesses, vulnerabilities and security risks in i4Q Solutions, to protect data, and to ensure and manage its security against attacks from internal and external sources.

1.1 Methodology for Test Cases

With the following test case planning and execution template, i4Q Solution Providers will be able to plan the testing of their Solutions, taking into account the individual components to be tested, execute the tests and analyze the resulting test data. The tests are identified by a test ID and name, test execution steps, priority levels and notes are also included, and finally by comparing the expected results with the actual results, a final conclusion is given.

i4Q **Solution**	< i4Q Solution>
Test Case **ID**	<unique id to identify the test case>
Test Case **Type**	<types of test cases>
Test Case **Description**	<brief summary of this test case>
Date Tested	<dd/mm/yyyy>
Module Stations **Involved**	<module station name>
Test Case **(Pass/Fail/Not Executed)**	<Pass/Fail/Not Executed >

Step #	**Prerequisites**		Step #	Test **Data**
1	<precondition description>		1	<necessary test data description>
2			2	
3			3	

Test **Scenario**	<long complementary test case description>

Step #	Step **Details**	**Results**		**Pass / Fail / Not executed**
		Expected	**Actual**	
1	<brief details of each step>	<brief explanation of the expected results of each step>	<brief explanation of the actual result of each step>	
2				
3				

Table 1. GENERIC Pilot Test Case Template

The information contained in the Test Case (Table 1) template is as follows:

- Section **1** – <u>Test Case Identification</u>:
  - **i4Q Solution**: Enter the i4Q code of the Test to be performed:
    - i4Q^{BC} - Blockchain Traceability of Data
    - i4Q^{TN} - Trusted Networks with Wireless & Wired Industrial Interfaces
    - i4Q^{DR} - Data Repository
    - i4Q^{DIT} -Data Integration and Transformation Services
    - i4Q^{DA} - Services for Data Analytics
    - i4Q^{AD} - Analytics Dashboard
    - i4Q^{IM} - Infrastructure Monitoring
    - i4Q^{DT}- Digital Twin simulation services
    - i4Q^{PQ} - Data-driven Continuous Process Qualification
    - i4Q^{QD} - Rapid Quality Diagnosis
    - i4Q^{PA} - Prescriptive Analysis Tools
    - i4Q^{LRT} - Manufacturing Line Reconfiguration Toolkit
  - Test Case **ID**: Unique alphanumeric identifier of the test case in the following format: " i4Q SolutionsCode"_TC-"nn". For example: **i4QBC_TC-03.**
  - Test Case **Type**: type of test case that will help to understand the purpose of each test case selecting one of the following types:
    - **Functional**
    - **User Interface**
    - **Performance**
    - **Integration**
    - **Usability**
    - **Database**
    - **Security**
  - Test Case **Description**: Summary of each test case.
  - **Date Tested**: Date of test execution with format <**dd/mm/yyyy**>
  - **Module Stations Involved**: Code of the module station(s) involved in the test, if none is used in the test enter "**NONE**", for the rest select from the following list:
    - **CP-AM-OUT** - Output
    - **CP-AM-iDRILL** - iDrilling
    - **CP-AM-CAM** - Camera inspection
    - **CP-AM-MAGBACK-BLACK** - Magazine
    - **CP-AM-MPRESS** - Muscle press
    - **CP-F-FBRANCHCP** - Factory Branch
    - **CP-F-ASRS32-P** - CP Factory High-bay storage for pallets
    - **CP-L-BRANCH** - CP Lab Branch
    - **Robotino** – Robotino
  - Test Case (**Pass/Fail/Not Executed**): Final status of the Test Case.

- Section **2 –** Pre-requisites and Test Data:
  - **Step #** and **Prerequisites**: Prerequisite Number and description.
  - **Step #** and **Test Data**: Test Data Number and description.

- Section **3** - Step Detail and Results:
  - Test **Scenario**: Long complementary test case description.
  - **Step #** and **Step Details**: step number and brief details of each step.
  - **Results - Expected** and **Actual**: Test case results:
    - **Expected**: brief explanation of the expected results of each step
    - **Actual**: brief explanation of the actual outcome of each step
  - **Pass / Fail / Not executed**: Final status of execution of each step.

In all those cases in which the definition of the Test Case or its results are not obtained until the Test Case has been carried out, it is indicated by the acronym "**TBD**" with the meaning "**To Be Defined**".

The final objective of each Test Case is to test, analyze or verify the execution behaviour of the i4Q Solutions under very specific parameters, always aiming at improving the Solution. The specific test execution parameters are chosen by i4Q Solution Providers based on their expert technical knowledge of their Solution and always thinking about on which i4Q Pilot could be verified in the best way.

2. Experimentation Facility - General Pipeline

The Generic pilot aims to experiment with the FESTO plant provided by the UPV (Polytechnic University of Valencia), briefly describing the different modules that make up the line for use in a smart factory environment for teaching and research purposes. The focus is not to improve the quality of the manufactured product, as the machines are designed for educational purposes, but to demonstrate how these i4Q solutions can be implemented in different environments and work with different types of data compared to previous pilots.

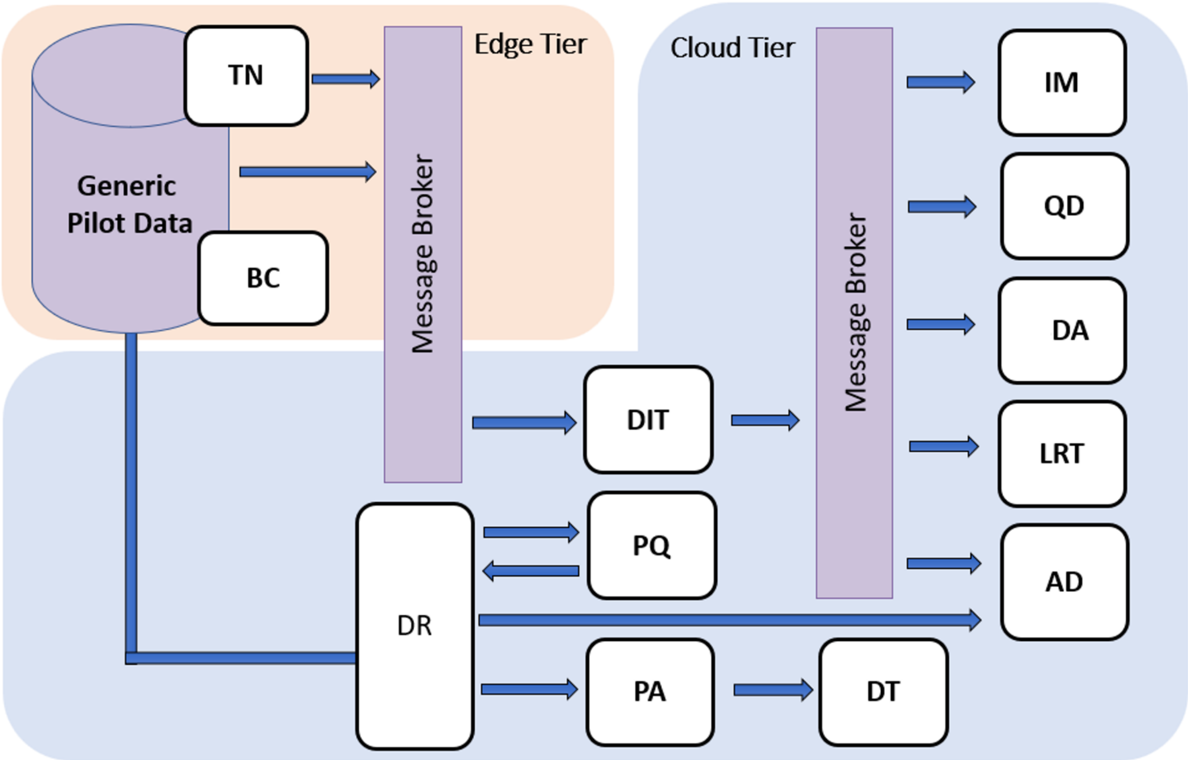

Figure 2. GENERIC Pilot Pipeline

As we can see in the generic pipeline (**Figure 2**), the intention is to be able to transport data from the different modules of the FESTO plant to the different solutions. For this purpose, two types of data clients have been implemented. On the one hand, the message broker for short-term data and, on the other hand, the i4Q^{DR} solution with a MongoDB for long-term data persistence. The other i4Q solutions will demonstrate that they can connect and interact with these data sources.

2.1 Individual Module Stations

The aim of this point is to explain all the elements that change when a request for the creation of a telephone is launched through the MES on FESTO machines. Each station of the module is equipped with Programmable Logic Controllers (PLC) with TCP connectivity (Fieldbus communication).

2.1.1 Physical (sensing)

Based on the search results, there is no direct information on specifying different workstations by looking at the data obtained by the PLCs. However, we can gather some information on PLCs that can be useful in this context.

A Programmable Logic Controller (PLC) is a small, sturdy industrial computer designed to control automated industrial processes and machines. PLCs are used almost everywhere, and they are the control hubs for various applications. The CPU of the PLC takes information about the inputs and performs logic on them to operate the output logic. The programming characteristics of the PLC should give the most efficient and effective control of the process.

In specifying different workstations, we can use PLCs to gather data from different parts of the manufacturing process and use that data to optimize the production process. For example, we can use PLCs to monitor the performance of different machines and workstations and gather data on their efficiency and productivity. This data can then be used to optimize the production process by identifying bottlenecks and areas for improvement.

CP-AM-CAM - Camera inspection	
Station	Description Important Values
	• **xBG1**: Indicates whether the Cam station is operating (*True*) or not (*False*). • **xBG1_BCD0**: Indicates whether the Cam station is operating (*True*) or not (*False*). • **xBG5**: Indicates whether the part is entering the Cam station (*True*) or not entering (False). • **xBG6**: Indicates whether the part is exiting the Cam station (*True*) or not passing through the exit (*False*). • **xG1_BG7_KG1**: No relevant. • **xG1_BG8_KG2**: No relevant. • **xG1_BG9:** When set to True it indicates that the Cam station operation is complete. When set to false it is waiting to know the final status of the station. • **xMB1:** When set to *True* it indicates that the Cam station operation is complete. When set to *false* it is waiting to know the final status of the station. • **xQA1_A1:** When it is set to *True* it means that the tape is in motion. If it is set to *False*, the tape is stopped.

Time (Sec)		1	2	3	4	5	6	7	8	9	10	11	12	13
Type	Sensor	State												
Input	xBG1	F					T				F			
	xBG1_BCD0	F					T				F			
	xBG5	F	T	F										
	xBG6	F											T	F
	xG1_BG7_KG1	F	T	F	T	F	T	F		T		F		
	xG1_BG8_KG2	T		F	T	F	T		F		T	F	T	
	xG1_BG9	F									T	F		
Output	xMB1	F									T	F		
	xQA1_A1	T					F			T				
		MOVE					Photo				MOVE			

Table 2. Module Station – CP-AM-CAM (FESTO, 2022e)

CP-AM-iDRILL - iDrilling	
Station	Description Important Values
	• **xBG1**: Indicates whether the iDrill station is operating (*True*) or not (*False*). • **xBG1_BCD0**: Indicates whether the iDrill station is operating (*True*) or not (*False*). • **xBG5**: Indicates whether the part is entering the iDrill station (*True*) or not entering (False). • **xBG6**: Indicates whether the part is exiting the iDrill station (*True*) or not passing through the exit (*False*). • **xG1_BG7_KG1**: No relevant. • **xG1_BG8_KG2**: No relevant. • **xG1_BG9:** When set to True it indicates that the iDrill station operation is complete. When set to false it is waiting to know the final status of the station. • **xMB1:** When set to *True* it indicates that the iDrill station operation is complete. When set to *false* it is waiting to know the final status of the station. • **xQA1_A1:** When it is set to *True* it means that the tape is in motion. If it is set to *False*, the tape is stopped.

Time (Sec)		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Type	Sensor	State																			
Input	xBG1	F				T											F				
	xBG1_BCD0	F				T											F				
	xBG5	F	T		F																
	xBG6	F																	T		
	xG1_BG7_KG1	F	T	F		T											F	T	F	T	
	xG1_BG8_KG2	T			F		T											F			
	xG1_BG9	F														T		F			
Output	xMB1	F														T		F			
	xQA1	F	T					F								T					
		MOVE				DRILL											MOVE				

Table 3. Module Station – CP-AM-iDRILL (FESTO, 2022f)

Robotino	
Station	Description Important Values
	• **xBG1**: Indicates whether the Cam station is operating (*True*) or not (*False*). • **xBG1_BCD0**: Indicates whether the Cam station is operating (*True*) or not (*False*). • **xBG5**: Indicates whether the part is entering the Cam station (*True*) or not entering (False).

Time (Sec)		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37
Type	Sensor	State																																					
Input	xBG1	F	T																														F				
	xBG1_BCD0	F	T																														F				
	xBG2	F					T				F																						T				
	xBG3	F					T				F																						T				
	xBG5	F								T																								T				F	
		Positioning					Take Part				Move Part																				Positioning					Give Part			

Table 4. Module Station – Robotino (FESTO, 2022c)

CP-AM-MAGBACK-BLACK - Magazine	
Station	Description Important Values
	• **xBG1**: Indicates whether the Magback station is operating (*True*) or not (*False*). • **xBG1_BCD0**: Indicates whether the Magback station is operating (*True*) or not (*False*). • **xBG5**: Indicates whether the part is entering the Magback station (*True*) or not entering (False). • **xBG6**: Indicates whether the part is exiting the Magback station (*True*) or not passing through the exit (*False*). • **xG1_BG9:** When set to True it indicates that the Magback station operation is complete. When set to false it is waiting to know the final status of the station. • **xMB1:** When set to *True* it indicates that the Magback station operation is complete. When set to *false* it is waiting to know the final status of the station. • **xQA1_A1:** When it is set to *True* it means that the tape is in motion. If it is set to *False*, the tape is stopped.

Time (Sec)		1	2	3	4	5	6	7	8	9	10	11	12	13	14
Type	Sensor	State													
Input	xBG1	F					T			F					
	xBG1_BCD0	F					T			F					
	xBG5	F	T		F										
	xBG6	F											T		
	xCL_BG1	T						F		T					
	xCL_BG2	F						T		F					
	xCL_BG3	T						F		T					
	xCL_BG4	F						T		F					
	xCL_BG5	T						F		T					
	xCL_BG7	F					T				F				
	xCL_BG8	F						T			F				
	xG1_BG7_KG1	T				F					T	F	T		
	xG1_BG8_KG2	T		F	T		F	T					F	T	
	xG1_BG9	F								T	F				
Output	XCL_MB1	T						F	T						
	XCL_MB3	T						F	T						
	XCL_MB4	F						T	F						
	XCL_MB5	F						T	F						
	xMB1	F								T	F				
	xQA1_A1	T						F	T						
		MOVE					Put Back				MOVE				

Table 5. Module Station – CP-AM-MAGBACK-BLACK (FESTO, 2022g)

CP-AM-MPRESS - Muscle press	
Station	Description Important Values
	• **xBG1**: Indicates whether the Mpress station is operating (*True*) or not (*False*). • **xBG1_BCD0**: Indicates whether the Mpress station is operating (*True*) or not (*False*). • **xBG5**: Indicates whether the part is entering the Mpress station (*True*) or not entering (False). • **xBG6**: Indicates whether the part is exiting the Mpress station (*True*) or not passing through the exit (*False*). • **xG1_BG9:** When set to True it indicates that the Mpress station operation is complete. When set to false it is waiting to know the final status of the station. • **xMB1:** When set to *True* it indicates that the Mpress station operation is complete. When set to *false* it is waiting to know the final status of the station. • **xQA1_A1:** When it is set to *True* it means that the tape is in motion. If it is set to *False*, the tape is stopped.

Time (Sec)		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Type	Sensor	State																		
Input	xBG1	F				T											F			
	xBG1_BCD0	F				T											F			
	xBG5	F	T	F																
	xBG6	F																T	F	
	xG1_BG7_KG1	T	F	T		F										T		F		T
	xG1_BG8_KG2	F	T			F										T	F		T	F
	xG1_BG9	F														T	F			
	xHBG1	T															F		T	
	xH_DQ3	F					T										F			
	wAnalogIn0	148						416	779	1039	1039	1039	1039	1039	1039	144				
	wAnalogIn1	717				969	1689	2266	2643	2747	2747	2747	2747	2747	2747	741				
Output	xMB1	F														T	F			
	xQA1_A1	T					F									T				
		MOVE				Press											MOVE			

Table 6. Module Station – CP-AM-MPRESS (FESTO, 2022h)

CP-AM-OUT - Output	
Station	Description Important Values
	• **xBG1**: Indicates whether the OutPut station is operating (*True*) or not (*False*). • **xBG1_BVD0**: Indicates whether the OutPut station is operating (*True*) or not (*False*). • **xBG5**: Indicates whether the part is entering the OutPut station (*True*) or not entering (False). • **xBG6**: Indicates whether the part is exiting the OutPut station (*True*) or not passing through the exit (*False*). • **xG1_BG9:** When set to True it indicates that the OutPut station operation is complete. When set to false it is waiting to know the final status of the station. • **xMB1:** When set to *True* it indicates that the OutPut station operation is complete. When set to *false* it is waiting to know the final status of the station. • **xQA1_A1:** When it is set to *True* it means that the tape is in motion. If it is set to *False*, the tape is stopped.

Time (Sec)		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Type	Sensor	State																	
Input	xBG1	F						T							F				
	xBG1_BCD0	F						T							F				
	xBG1_BCD3	F			T	F													
	xBG5	F	T	F															
	xBG6	F															T		F
	xG1_BG7_KG1	F		T		F	T									F	T	F	
	xG1_BG8_KG2	F	T			F		T						F	T	F		T	
	xG1_BG9	F													T	F			
	xGM_BG1	T							F		T								
	xGM_BG2	F							T	F									
	xGM_BG3	T							F			T							
	xKf1_DO0	T						F	T		F	T	F	T					
	xKF1_DO10	F					T		F		T				F				
Output	xGM_MB1	T							F	T									
	xGM_MB4	T								F	T								
	xKF1_DI1	F					T			F	T								
	xKF1_DI10	F					T		F		T	F							
	xKF1_DI2	F					T			F	T								
	xKF_DI6	F					T		F		T	F							
	xMB1	F														T	F		
	xQA1_A1	F	T		F														
		MOVE					Save the phone									MOVE			

Table 7. Module Station – CP-AM-OUT (FESTO, 2022i)

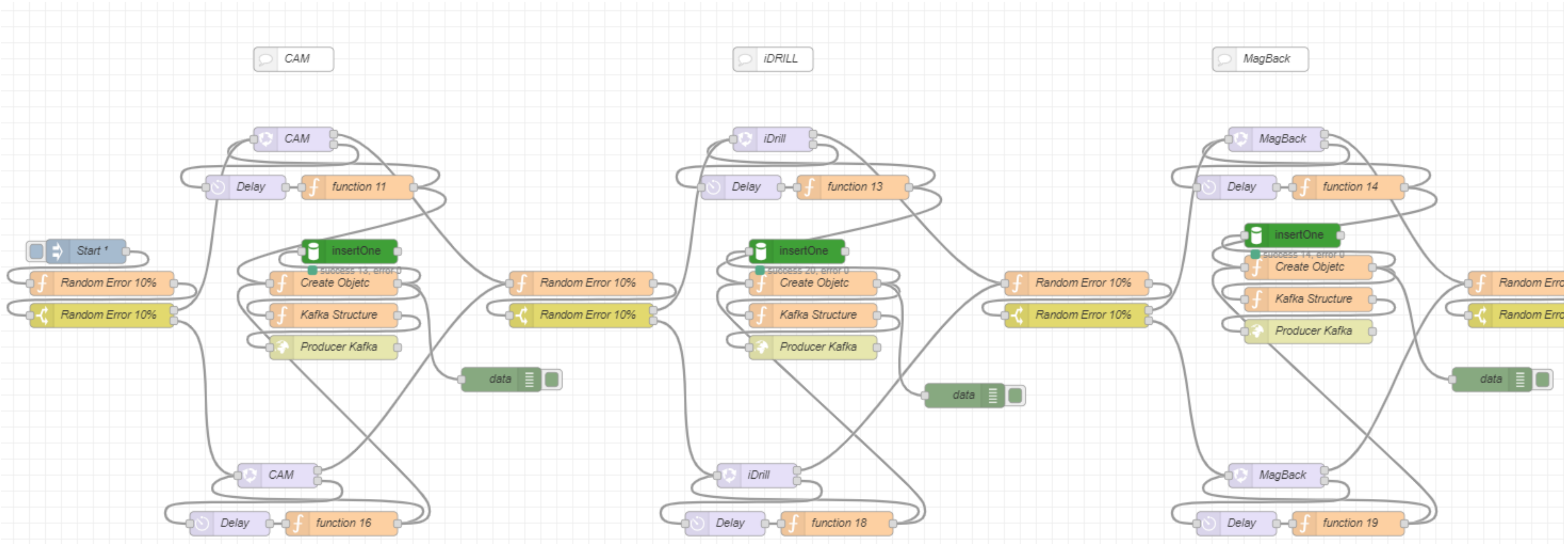

Figure 3. Simulator architecture

The simulator has been developed to facilitate access to the different partners for the use of the Generic Pilot. The reason for its development is due to the difficulty of access by users to the plant and so that they can test in a safe environment. This simulator is accessible from the URL https://reverse.i4q.cigip.upv.es

2.1.2 Data from Stations

The implementation architecture described in the generic pilot involves acquiring data from different FESTO stations through the OPC UA protocol and managing this data using the Node-RED solution. Subsequently, the data is stored in two different locations: a Kafka system and the i4Q[DR] solution, which will deploy MongoDB as a database and MinIO as a high-performance distributed object storage. The goal is for the rest of the i4Q solutions to interact with this data stored in Kafka and i4Q[DR].

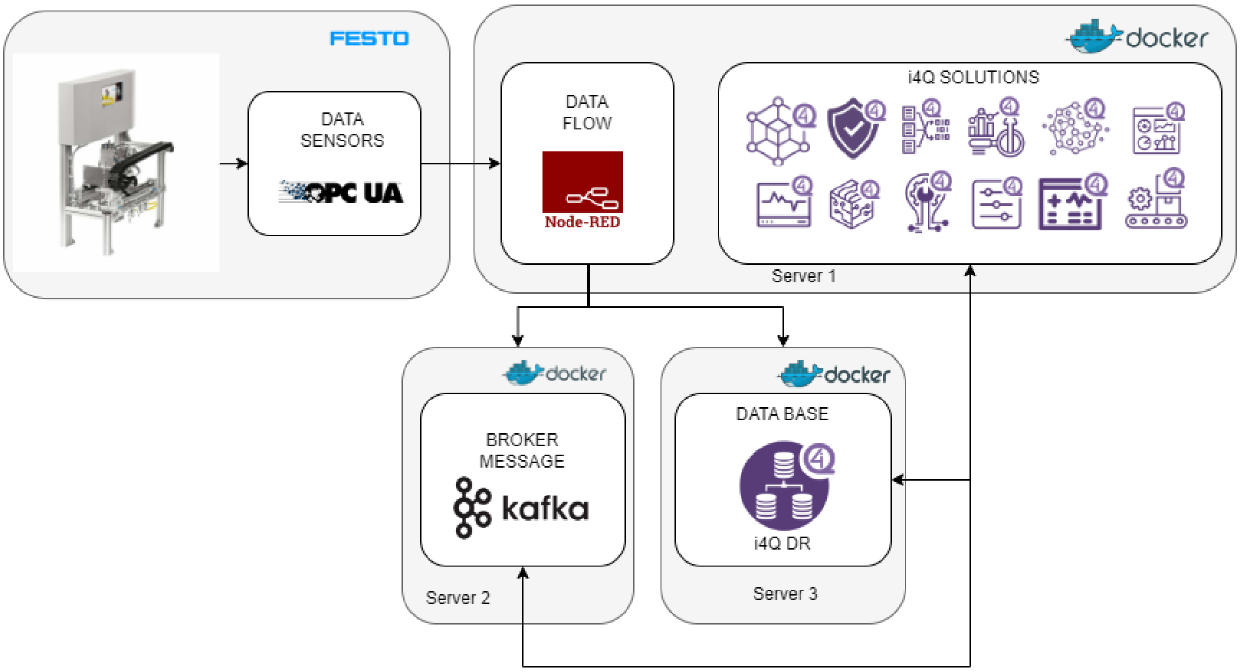

Figure 4. Data Flow FESTO Machine

Therefore, it will operate as follows:

- **Data Acquisition from FESTO Stations:** The FESTO stations send data using the OPC UA protocol. OPC UA is a standard communication protocol used in the industry to exchange data between different systems in real time. The FESTO stations must be configured to expose their data via OPC UA.
- **Node-RED as Middleware:** Node-RED handles data acquisition and subsequent processing. Node-RED is a visual programming environment that simplifies the creation of data flows and automation. In Node-RED, specific nodes are configured to connect to the FESTO stations through OPC UA and perform any necessary data processing on the received data.
- **Data Storage in Kafka:** The processed data in Node-RED is sent to a Kafka system. Kafka is a data streaming platform that enables real-time data ingestion and subsequent processing and analysis. The data is written to a specific topic in Kafka, allowing other solutions to consume it efficiently.

- **Data Storage in i4Q^{DR} (MongoDB):** Besides Kafka, the data is also stored in i4Q^{DR}, which uses MongoDB as a database. *MongoDB* is a NoSQL database that is scalable and suitable for storing large amounts of unstructured or semi-structured data. The data is organized and stored in collections within MongoDB, enabling efficient access for queries and further analysis.
- **Interaction with Stored Data:** The other i4Q solutions can interact with the data stored in Kafka and i4Q^{DR} according to their needs. They can consume data from Kafka using Kafka subscribers and query data in MongoDB using the appropriate queries.

2.1.3 Network

Each solution is accompanied by the final URL where it has been implemented. It should be noted that all these solutions have been deployed together on the same server using Docker images. In addition, a proxy based on Nginx with a certificate issued by Let's Encrypt has been used to guarantee the security and performance of these implementations. Using a Nginx proxy with Docker and Let's Encrypt certificate allows for efficient handling of incoming requests, providing additional security by encrypting communication using HTTPS. This configuration ensures a secure and reliable environment for the solutions deployed on the shared server.

3. Test Cases - i4Q Solutions Testing and analysis of results

3.1 Test Case of i4Q^{BC} - Blockchain Traceability of Data

3.1.1 Test prerequisites, data and step details

In this section, the main tests carried out to validate the implementation of the i4Q Blockchain Traceability of Data solution in the Generic Pilot are presented.

i4Q **Solution**	i4Q Blockchain Traceability of Data
Test Case **ID**	i4QBC_TC-01
Test Case **Type**	User Acceptance Test
Test Case **Description**	Track the historical machine configuration, and serve as a single point of truth, preserving provenance and supporting non-repudiation for the machine configurations
Date Tested	TBD
Module Stations **Involved**	TBD
Test Case **(Pass/Fail/Not Executed)**	TBD

Step #	**Prerequisites**		Step #	Test **Data**
1	Deploy and integrate the solution in the pilot		1	N/A
2	Setup users and machines with the administration user interface		2	User data, machine data, and assignment of responsible users for the machines
3	Select a machine for test		3	N/A

Test **Scenario**	In this scenario, the quality engineer, line manager, and producers of machine can play, respectively, the role operator, controller, and auditor. A quality manager (operator) proposes a change to the configuration of the machine. A line manager (controller) checks if the configuration is valid and should be applied to the machine and make decisions on approval or dis-approval. In case the configuration is applicable, then the change request of configuration is approved and saved in the Blockchain (Orion server). Later, the users, such as, machine producer or production manager, takes the role as auditor to review & audit the historical changes of the machine configuration. They can, for example, use the records to check the compliance of the machine usage, or link the machine

	configuration data with product quality data or warranty & maintenance cases to help data-driven decision making.

Step #	Step Details	Results		Pass / Fail / Not executed
		Expected	**Actual**	
1	Quality manager initialize machine configuration for the selected machine	Initial machine configuration is submitted and wait for approval from the line manager	TBD	TBD
2	Line manager approve the initialized machine configuration	Machine configuration is initialized	TBD	TBD
3	Quality manager request to update the machine configuration for the production of a new batch of products	Update machine configuration is awaiting approval from the line manager	TBD	TBD
4	Line manager check the update request from the quality manager, and find the configuration is not optimal. She / He, thereafter, reject the update request.	Update request got rejected	TBD	TBD
5	Quality manager suggest a new set of machine configuration, and submit a new update request.	Update request is awaiting approval from line manager	TBD	TBD
6	Line manager re-check the update request, and approve the new suggested machine configuration	Update request is approved	TBD	TBD
7	Production manager reviews the historical machine configuration to assess compliance & guarantee rules	Product manager can audit the historical records of the machine configuration and see the blockchain transactions	TBD	TBD

Table 8. Test Case i4QBC_TC-01

3.1.2 Results and conclusions

The i4Q Blockchain Traceability of Data solution is able to takes advantage of the Blockchain (BC) technology, i.e., Hyperledger Orion blockchain database, to track the value changes of the machine configuration data and provide the capability of tracking data validation and verification as well as data auditing. It demonstrates how operators, controllers, and auditors can use the i4Q^BC Solution to manage and track the machine configuration data.

3.2 Test Case of i4Q^TN - Trusted Networks with Wireless & Wired Industrial Interfaces

3.2.1 Test prerequisites, data and step details

This section summarizes the proposed Test Cases to validate the different subcomponents implemented of the i4Q Trusted Networks with Wireless & Wired Industrial Interfaces solution, deployed and partial integrated in the Generic Pilot. The following Test Cases should be considered independently for the wired and wireless part, and from now on, the identification IWSN for wireless and TSN for wired will be added to the solution name.

i4Q **Solution**	i4Q^TN - Trusted Networks with Wireless & Wired Industrial Interfaces - IWSN
Test Case **ID**	i4QTN_TC-01
Test Case **Type**	Usability
Test Case **Description**	Starting sensor data collection
Date Tested	TBD
Module Stations **Involved**	iDrilling (CP-AM-iDRILL) or Robotino
Test Case **(Pass/Fail/Not Executed)**	TBD

Step #	**Prerequisites**		Step #	Test **Data**
1	Have the following physical elements of the solution: IWSN Gateway, at least one IWSN node, at least one compatible sensor.		1	N/A
2	Select the proper location for the IWSN Gateway, which needs a power supply and ethernet connection to the i4Q main broker.		2	N/A
3	Prior the wireless nodes deployment, the gateway should be already turned on.		3	Once the gateway is deployed, logging to the solution user interface to validate the proper functionality.

Test **Scenario**	The aim of this Test case is to validate the system deployment, configuration and integration in a generic scenario, being able to locate the wireless nodes at fixed stations such as the iDrilling or even at mobile stations like the Robotino. The Test Case guide the user over the deployment and configuration of the wireless node and the connection and parametrization of compatible industrial sensors. The new source of data acquisition will be available through the common i4Q broker. To guarantee the integrity of the FESTO demonstrator, the industrial sensor provided will be non-intrusive, giving environmental and local information around the Generic Pilot stations.

Step #	Step **Details**	**Results**		**Pass / Fail / Not executed**
		Expected	**Actual**	
1	Select the location of the wireless node to be deployed, for example the iDrilling station or the Robotino.	At fixed stations the node could be powered by a power supply and at mobile stations the battery power is mandatory.	TBC	TBC
2	Turn on the wireless node using the power on switch of the node.	The led interface shows the network connectivity.	TBC	TBC
3	Connection validation to the gateway.	The user interface shows connectivity and quality parameters of the node link.	TBC	TBC
4	Configuration of new sensor acquisition.	New data sources collected from the node.	TBC	TBC
5	Visualization and integration of new data flows.	The new information collected is represented in the user interface and integrated to the common i4Q architecture.	TBC	TBC

Table 9. Test Case i4QTN_TC-01

i4Q **Solution**	i4Q^{TN} - Trusted Networks with Wireless & Wired Industrial Interfaces - TSN
Test Case **ID**	i4QTN_TC-02
Test Case **Type**	Usability
Test Case **Description**	Configuring TSN network
Date Tested	TBD
Module Stations **Involved**	NONE
Test Case **(Pass/Fail/Not Executed)**	TBD

Step #	**Prerequisites**		Step #	Test **Data**
1	Have a physical TSN network set up, containing at least one switch and two end systems, both capable of handling TSN messages.		1	N/A
2	Make sure that the switches and end systems are included into the configuration tool		2	N/A
3	Define messages to be sent over the TSN network and what are the timing of the said messages.		3	No specific test data required. Most of the time just random data where different data lengths (e.g., min and max) are send. Content is not of importance.

Test **Scenario**	The aim of this test case is to validate that a correct schedule is generated for sending the defined messages over the TSN network. This setup is not deployed in any station of the generic demonstrator, as we were not allowed to modify the demonstrator, which was required to integrate the TSN solution. Therefore, the test case will be performed in our own labs with an own demonstrator. The test case will guide the user for setting up a TSN network and specifying the correct timing of the messages in the network.

Step #	Step **Details**	**Results**		**Pass / Fail / Not executed**
		Expected	**Actual**	
1	Select the required switches and end systems and drag them on the topology planner	Switches and end systems positioned in the topology editor	TBC	TBC

2	Define connections between end systems and switches	Connections drawn between end systems and switches within the topology editor	TBC	TBC
3	Parameterization of connections	The user interface enables the developer to parameterize the connection and streams between the different nodes in the network	TBC	TBC
4	Schedule generation	A network schedule is generated based on the parameterization of the nodes and connections. It can be viewed in the tool.	TBC	TBC
5	Deploy schedule on hardware	If the network schedule is correct, it can be deployed on the real hardware.	TBC	TBC

Table 10. Test Case i4QTN_TC-02

3.2.2 Results and conclusions

The execution of the Test Cases, related to the i4QTN solution, allows to exemplify different use cases of both subcomponents of the solution, the wired and the wireless parts. These scenarios guide the user through the deployment and configuration until the fully deployed system. Regarding the IWSN Test Case, the use of the IEEE 802.15.4e protocol, in addition to the Time Sensitive Channel Hopping medium access, guarantees the robust communication exchange, and in combination with the SDN controller allows a deterministic behaviour creating the mesh network and scheduling the messages exchanged based on the available radio resources. The other test case, targeting the IEEE 802.1 TSN solution, guarantees deterministic wired communication. The test case shows how to configure and generate a network schedule in a network tool, which can then finally be deployed on the real hardware.

3.3 Test Case of i4Q^{DR} - Data Repository

3.3.1 Test prerequisites, data and step details

In this section, the main tests carried out to validate the implementation of the i4Q Data Repository solution in the Generic Pilot are presented.

i4Q **Solution**	i4Q Data Repository
Test Case **ID**	i4QDR_TC-01
Test Case **Type**	Solution configuration and deployment
Test Case **Description**	The objective of this test case is to verify that the configuration and deployment of the corresponding Docker containers for the data storage technologies (e.g. MongoDB, MySQL, Redis, MinIO, etc.) and the scenario specified by the user can be performed correctly, as well as to verify that the configuration and deployment of Trino has been performed correctly so that it can work with the selected technology.
Date Tested	03/01/2024
Module Stations **Involved**	CP-AM-OUT, CP-AM-iDRILL, CP-AM-CAM, CP-AM-MAG, CP-AM-MPRESS, CP-F-ASR32-P, and CP-L-BRANCH.
Test Case **(Pass/Fail/Not Executed)**	Pass

Step #	**Prerequisites**		Step #	Test **Data**
1	Install the software dependencies required.			N/A
2	Create the private keys and SSL certificates by using the ssl/_prepare.sh script.		1	$ ssl/_prepare.sh
3	Execute the run.sh script to prepare the Docker Compose YAML files with the appropriate configuration. Then, it deploys the MongoDB and Trino technology based on these files.		2	$ ./run.sh --use mongodb:tls

Test **Scenario**	The Data Repository solution allows you to configure, launch and manage a set of tools and technologies related to data storage and management. These include standard products such as MongoDB, MySQL, Redis, MinIO, etc. The aim of these toolkits is to provide automatable mechanisms for configuring, deploying, managing, diagnosing and disposing the tools and technologies. Each tool or technology supports different scenarios. For example, the MongoDB tool offers the "basic" scenario to deploy a single MongoDB server without any other extended features, the "repltls" scenario for to deploy a replica set of MongoDB servers with TLS support, etc. The aim of this test scenario is to verify that the configuration and deployment of the MongoDB toolkit for the scenario using TLS certificates has been done correctly, as well as to verify that the configuration and deployment of Trino has been done correctly to work with this technology.

Step #	Step **Details**	**Results**		**Pass / Fail / Not executed**
		Expected	**Actual**	
1	Create the SSL certificates.	SSL certificates are expected to be created.	SSL certificates have been successfully created.	Pass
2	Prepare the Docker Compose YAML files with the appropriate configuration.	Docker Compose YAML files are expected to be created.	Docker Compose YAML files have been successfully created.	Pass
3	Deploy the MongoDB Docker container using the YAML file prepared.	MongoDB server with TLS certificate support is expected to be deployed in a Docker container.	Docker container with the MongoDB server supporting TLS certificates has been successfully deployed.	Pass
4	Deploy the Trino Docker container using the YAML file prepared.	It is expected to deploy a Docker container with the Trino tool from the YAML file provided.	Docker container with the Trino tool has been successfully deployed from the YAML file provided.	Pass

Table 11. Test Case i4QDR_TC-01

3.3.2 Results and conclusions

After executing the test case defined in the previous subsection, it can be concluded that the solution Data Repository is capable of configuring, deploying, managing, diagnosing and disposing a set of storage and data management tools and technologies, adapting to the different user needs by the definition of different scenarios. All this is done through the use of Docker Compose YAML files, which allow to deploy the different storage and data management technologies in the form of Docker containers, achieving a fast, simple and portable deployment.

3.4 Test Case of i4Q^{DIT} - Data Integration and Transformation Services

3.4.1 Test prerequisites, data and step details

This section presents the usability and implementation of DIT in the generic pilot.

i4Q **Solution**	i4Q^{DIT} - Data Integration and Transformation Services
Test Case **ID**	i4QDIT_TC-01
Test Case **Type**	Functional
Test Case **Description**	To draw data from the sensors used for conveyor detection, apply some pre-processing functions and extract some features and then send the created data to another solution for analysis (e.g. via message broker).
Date Tested	TBD
Module Stations **Involved**	CP-L-BRANCH - CP Lab Branch
Test Case **(Pass/Fail/Not Executed)**	Pass

Step #	**Prerequisites**		Step #	Test **Data**
1	N/A		1	N/A
2	N/A		2	N/A
3	N/A		3	N/A

Test **Scenario**	<long complementary test case description>

Step #	Step **Details**	**Results**		**Pass / Fail / Not executed**
		Expected	**Actual**	
1	Draw data from DR	TBD	TBD	TBD
2	Apply preprocessing function	TBD	TBD	TBD
3	Send the data through message broker	TBD	TBD	TBD

Table 12. Test Case i4QDIT_TC-01

3.4.2 Results and conclusions

Results and conclusions will be established at the time of the test case execution.

3.5 Test Case of i4Q^{DA} - Services for Data Analytics

3.5.1 Test prerequisites, data and step details

In this section, the main test carried out to validate the implementation of the i4Q Data Analytics solution in the Generic Pilot is presented:

i4Q **Solution**	i4Q^{DA} - Services for Data Analytics
Test Case **ID**	i4QDA_TC-01
Test Case **Type**	Functional
Test Case **Description**	Analysis of the quality of the piece related with the force applied to it in the CP-AM-MPRESS station
Date Tested	22/01/2024
Module Stations **Involved**	CP-AM-MPRESS - Muscle press
Test Case **(Pass/Fail/Not Executed)**	TBD

Step #	**Prerequisites**		Step #	Test **Data**
1	Definition of the type of machine learning analysis to be executed over the data from the station		1	Force generated by the press station on the piece
2	Development of the workflow for the execution of the ML analysis		2	Quality of the piece related of whether the part is a good part or a bad part
3	N/A		3	N/A

Test **Scenario**	The aim of this test scenario is to test the data analytics solution regarding its usability for the analysis of a process from the pipeline of the FESTO plant. By gathering data from the CP-AM-PRESS station regarding the force applied over the part that goes to that station, and by collecting data related to the quality of the piece, i.e., if the piece is usable (good part) or if is a piece with a defect and cannot be used (bad part) for that same piece, it is possible to create, using the data analytics tool, a workflow for the analysis of the quality of the end product based of the force generated in the station, which then could be used for forecasting the end result of the process.

Step #	Step Details	Results		Pass / Fail / Not executed
		Expected	Actual	
1	Train model for prediction of quality using the workflow creator	Model is created and the training presents stats that validate the model	TBD	TBD
2	Predict the quality of the parts that go through the press	Prediction of the quality based on the data from the press, identifying correct defects	TBD	TBD

Table 13. Test Case i4QDA_TC-01

3.5.2 Results and conclusions

The execution of the Test Case, regarding the i4Q^{DA} solution, aims to show how this solution can be applied in the context of an industrial pipeline, by acting over a specific station that is present in the shopfloor. The execution of workflows to provide analytical insight over the status of the process aim to help in the decision-making regarding if there is a need to act to assure the correct operation of the station, since it can impact the outcome of the end product. The main focus of the test was regarding execution of the AI workflow and the showing of results.

3.6 Test Case of i4Q^{AD} - Analytics Dashboard

3.6.1 Test prerequisites, data and step details

In this section, the main test carried out to validate the implementation of the i4Q Analytics Dashboard solution in the Generic Pilot is presented:

i4Q **Solution**	i4Q^{AD} - Analytics Dashboard
Test Case **ID**	i4QAD_TC-01
Test Case **Type**	Functional
Test Case **Description**	Creation of a Dashboard with historical data stored in i4Q^{DR}, for the visualization of properties regarding the output from the CP-AM-OUT station
Date Tested	22/01/2024
Module Stations **Involved**	CP-AM-OUT - Output
Test Case **(Pass/Fail/Not Executed)**	TBD

Step #	**Prerequisites**		Step #	Test **Data**

1	N/A		1	Historical Data regarding part collection from the station
2	N/A		2	N/A
3	N/A		3	N/A

Test **Scenario**	The aim of this test case scenario is to test the creation of dashboards using this visualization tool, by connecting it to an existing data source, in this particular case is the i4Q^{DR}, and to extract historical data from the FESTO pipeline, where is this case it is the CP-AM-OUT station, in order to create visualizations, as in charts or tables depending on the type of data to be shown, and to aggregate those visualizations into one dashboard

Step #	Step **Details**	**Results**		**Pass / Fail / Not executed**
		Expected	**Actual**	
1	Connect the i4Q^{AD} to the i4Q^{DR}	A connection to the database which allows to access the data present	TBD	TBD
2	Selection of the data to be visualized	Creation a sub dataset with the data to be visualized using the tools functionalities	TBD	TBD
3	Creation of charts/tables	Creation of charts or tables by selecting the dataset created in the previous step	TBD	TBD

Table 14. Test Case i4QAD_TC-01

3.6.2 Results and conclusions

The execution of the Test Case, regarding the i4Q^{AD} solution, aims to show how this solution can be applied in the context of an industrial pipeline, regarding its usage to visualize historical data related to the production. This test case contains the steps necessary for the connection to the data source where the data is storage, as well as the selection of the data to be visualized, the selection of what types of visualization to be used, and the creation of a dashboard which contains all the visualizations created. The usage of this tool can help keep track of what is being produced, and to give some insights over all the data collected during the production process of the end product.

3.7 Test Case of i4Q^IM - Infrastructure Monitoring

3.7.1 Test prerequisites, data and step details

This section provides a description of the tests that have been carried out to validate the implementation and operation of the i4Q Infrastructure Monitoring solution in the Generic Pilot.

i4Q **Solution**	i4Q^IM - Infrastructure Monitoring
Test Case **ID**	i4QIM_TC-01
Test Case **Type**	Performance
Test Case **Description**	Condition monitoring of the pressure module of the FESTO plant.
Date Tested	TBD
Module Stations **Involved**	CP Muscle Press
Test Case **(Pass/Fail/Not Executed)**	TBD

Step #	**Prerequisites**		Step #	Test **Data**
1	Establish a secure connection with the i4Q^DR or the Message Broker using the appropriate generated certificates. This will allow for the ingestion of pre-processed (i4Q^DIT) and raw data generated from the machine.		1	N/A
2	Develop a predictive model or fine-tune a pre-trained one to detect potential failures of the machine.		2	Feature data from the muscle press obtained through the sensors of the module. Logs related to the module's status, to correlate the sensor data with the occurrence of a failure.

Test **Scenario**	The i4Q^IM allows for the exploitation of real-time machine sensor data to effectively detect the degradation of machine components or general machine failures. Through the usage of machine learning (ML) condition monitoring algorithms the solution can correlate the sensor data with historical problematic operation events and offer proactive alerts, to inform the operators to take corrective actions. The purpose of this test scenario is to check the compatibility of the i4Q^IM solution with the "CP Muscle Press" pressure

	module and to validate its effectiveness in predicting the occurrence of machine failures in during the pressing operation. After the execution of the solution, an alert is expected to be generated, informing the operator of a potential problem.

Step #	Step Details	Results		Pass / Fail / Not executed
		Expected	Actual	
1	Launch the solution and select a predictive model.	Data coming from the sensors are consumed through the Message Broker & real-time analytics begin.	TBD	TBD
2	Predict the status of the machine during a normal operation.	The sensor charts in the UI indicate the proper operation of the machine. Also, the prediction results are stored in the i4Q^{DR}.	TBD	TBD
3	Predict the status of the machine during a problematic operation.	The sensor charts and a notification element in the UI indicate the instances of a detected machine failure. Also, the prediction results are stored in the i4Q^{DR}.	TBD	TBD

Table 15. Test Case i4QIM_TC-01

3.7.2 Results and conclusions

Results and conclusions will be established at the time of the test case execution.

3.8 Test Case of i4Q^{DT} - Digital Twin Simulation Services

3.8.1 Test prerequisites, data and step details

In this section, the main tests carried out to validate the implementation of the i4Q Digital Twin solution in the Generic Pilot are presented.

i4Q **Solution**	i4Q^{DT} - Digital Twin
Test Case **ID**	i4QDT_TC-01
Test Case **Type**	Performance
Test Case **Description**	Variation of module processing time and verification of its impact in the overall final production rate of the FESTO plant
Date Tested	2024/01/04
Module Stations **Involved**	All of them, as the model tries to simulate the general behaviour of the pipeline
Test Case **(Pass/Fail/Not Executed)**	Pass

Step #	**Prerequisites**		Step #	Test **Data**
1	Summary of the processing time of all the stations involved in the module		1	Processing time of each station obtained from the node-network of the system
2	Development of the individual FMUs to simulate each of the module stations		2	Parameterization data of each station obtained from the documentation of each module
3	Development of the overall co-simulation model to simulate the whole FESTO plant		3	Parameterization of the overall pipeline and the relation between stations obtained from the description of the real system

Test **Scenario**	The aim of this test scenario is to test the model developed with the i4Q^{DT} solution for simulation of the pipeline of the FESTO plant. This model is built using the Physics-based workflow of the DT and is composed of several generic individual FMUs that simulate the behaviour of each module station. Once the co-simulation model is built, the validation of the model is carried out changing the parameters of each of those FMU models and verifying that the simulation results agree with the expected behaviour in the real FESTO plant. In this test scenario, the parameter to be evaluated is the processing time of the station, this is, the time that each

	station takes to process one product before continuing to the following station.

Step #	Step **Details**	**Results**		**Pass / Fail / Not executed**
		Expected	**Actual**	
1	Select station iDrilling and change its processing time	Simulated behaviour of the final production rate according to reality	The production rate decreases when the processing time increases	Pass
2	Select station Camera and change its processing time	Simulated behaviour of the final production rate according to reality	The production rate decreases when the processing time increases	Pass

Table 16. Test Case i4QDT_TC-01

i4Q **Solution**	i4Q[DT] - Digital Twin
Test Case **ID**	i4QDT_TC-02
Test Case **Type**	Performance
Test Case **Description**	Variation of the number of parallel processing modules and verification of its impact in the overall final production rate
Date Tested	2024/01/04
Module Stations **Involved**	All of them, as the model tries to simulate the general behaviour of the pipeline
Test Case **(Pass/Fail/Not Executed)**	Pass

Step #	**Prerequisites**		Step #	Test **Data**
1	Summary of the number of processing modules of all the stations involved in the module		1	Number of each processing modules of each station obtained from the node-network of the system
2	Development of the individual FMUs to simulate each of the module stations		2	Parameterization data of each station obtained from the documentation of each module
3	Development of the overall cosimulation model to simulate the whole FESTO plant		3	Parameterization of the overall pipeline and the relation between stations obtained from the description of the real system

Test **Scenario**	The aim of this test scenario is to test the model developed with the i4Q[DT] solution for simulation of the pipeline of the FESTO plant. This model is built using the Physics-based workflow of the DT and is composed of several generic individual FMUs that simulate the behaviour of each module station. Once the co-simulation model is built, the validation of the model is carried out changing the parameters of each of those FMU models and verifying that the simulation results agree with the expected behaviour in the real FESTO plant. In this test scenario, the parameter to be evaluated is the number of parallel processing modules of each station, this is, the number of products that could be processed simultaneously, according to the number of modules available in each station.

Step #	Step **Details**	**Results**		**Pass / Fail / Not executed**
		Expected	**Actual**	
1	Select station iDrilling and change the number of processing modules	Simulated behaviour of the final production rate according to reality	The production rate does not change when the number of processing modules of a certain module station increases	Pass
2	Select station Camera and change the number of processing modules	Simulated behaviour of the final production rate according to reality	The production rate does not change when the number of processing modules of a certain module station increases	Pass

Table 17. Test Case i4QDT_TC-02

3.8.2 Results and conclusions

The main result and conclusion of these tests is that the model is accurate and can, both quantitatively and qualitatively, simulate the behaviour of the real system.

3.9 Test Case of i4Q^{PQ} - Data-driven Continuous Process Qualification

3.9.1 Test prerequisites, data and step details

i4Q **Solution**	i4Q^{PQ} – Data-Driven Continuous Process Qualification
Test Case **ID**	i4QPQ_TC-01
Test Case **Type**	Performance
Test Case **Description**	Process quality evaluation in the drilling machine.
Date Tested	TBD
Module Stations **Involved**	CP-AM-iDRILL - iDrilling
Test Case **(Pass/Fail/Not Executed)**	TBD

Step #	**Prerequisites**		Step #	Test **Data**
1	Establish a secure connection with the i4Q^{DR} or the Message Broker using the appropriate generated certificates.		1	N/A
2	Measure quality criteria from workpiece and create a digitized array (csv.file or directly in i4Q^{DR} or Message Broker)		2	Univariate data of quality criteria

Test **Scenario**	The i4Q^{PQ} creates a comprehensive overview of a critical quality measurement, which can be evaluated either in a static or in a dynamic environment. In both, the Process Capability Indicator Cpk is calculated, and direct recommendations are drawn for the user. Within the test case, i4Q^{PQ} will be used to validate its interoperability and computing efficiency during the process. Conclusively, the quality output of the process can be monitored directly in the user interface next to the drilling machine.

Step #	Step **Details**	**Results**		**Pass / Fail / Not executed**
		Expected	**Actual**	
1	Select the details in the UI	User has to insert details in the UI like Tolerance Levels,	TBD	TBD

		data connection and forecasting window.		
2	Consume data from data source	Once the data can be consumed, an analysis of the underlying process performance will be drawn.	TBD	TBD
3	Interpret current process capability	For the last step it is important to combine the interpretation with implicit process knowledge to receive the highest information gain of the solution.	TBD	TBD

Table 18. Test Case i4QPQ_TC-01

3.9.2 Results and conclusions

Conclusions are that structural failures of the process can be minimized in advance due to the forecast of the process capability.

3.10 Test Case of i4Q[QD] - Rapid Quality Diagnosis

3.10.1 Test prerequisites, data and step details

i4Q **Solution**	i4Q[QD] – Rapid Quality Diagnosis			
Test Case **ID**	i4QQD_TC-01			
Test Case **Type**	Performance			
Test Case **Description**	Product quality control on the pressure module of the FESTO plant.			
Date Tested	TBD			
Module Stations **Involved**	CP Muscle Press			
Test Case **(Pass/Fail/Not Executed)**	TBD			
Step #	**Prerequisites**		Step #	Test **Data**
1	Establish a secure connection with the i4Q[DR] or the Message Broker		1	N/A

	using the appropriate generated certificates. This will allow for the ingestion of pre-processed (i4Q[DIT]) and raw data generated from the machine.			
2	Develop a predictive model or fine-tune a pre-trained one to predict the quality of the final product.		2	Feature data from the muscle press obtained through the sensors of the module. Logs containing past production quality records, to correlate the sensor data with the occurrence of a product defect.

Test **Scenario**	The i4Q[QD] allows for the exploitation of real-time machine sensor data to effectively detect non-optimal production condition or defects in the final product. The solution employs product quality conformity detection algorithms based on ML, to correlate machine sensor data deriving with prior problematic operation events and offer proactive alerts, to inform the operators to take corrective actions. The purpose of this test scenario is to check the compatibility of the i4Q[QD] solution with the "CP Muscle Press" pressure module and to validate its effectiveness in predicting the occurrence of defective products during the pressing of the phone cover. After the execution of the solution, an alert is expected to be generated, informing the operator of a potential problem.

Step #	Step **Details**	**Results**		**Pass / Fail / Not executed**
		Expected	**Actual**	
1	Launch the solution and select a predictive model.	Data coming from the sensors are consumed through the Message Broker & real-time analytics begin.	TBD	TBD
2	Predict the quality of the product during normal operation.	The sensor charts in the UI indicate the production of sound products. Also, the	TBD	TBD

		prediction results are stored in the i4Q^DR.		
3	Predict the quality of the product during a problematic operation.	The sensor charts and a notification element in the UI indicate defective production. Also, the prediction results are stored in the i4Q^DR.	TBD	TBD

Table 19. Test Case i4QQD_TC-01

3.10.2 Results and conclusions

Results and conclusions will be established at the time of the test case execution.

3.11 Test Case of i4Q^PA - Prescriptive Analysis Tools

3.11.1 Test prerequisites, data and step details

In this section, the main tests carried out to validate the implementation of the i4Q Prescriptive Analysis Tool solution in the Generic Pilot are presented.

i4Q **Solution**	i4Q^PA - Prescriptive Analysis Tool
Test Case **ID**	i4QPA_TC-01
Test Case **Type**	Usability
Test Case **Description**	The app is going to be tested by means of optimizing the manufacturing line's performance through the variation of the number of parallel processing modules and verification of its impact in the overall final production rate of the Festo plant
Date Tested	2024/01/08
Module Stations **Involved**	All of them, as the model tries to simulate the general behaviour of the pipeline
Test Case **(Pass/Fail/Not Executed)**	Pass

Step #	**Prerequisites**		Step #	Test **Data**
1	Obtain Festo plant's model created by i4Q^DT		1	N/A
2	Define model's parameters' range		2	N/A
3	Simulate model's variations		3	Simulations' results

4	Evaluate simulations results and prescribe minimum process time model		4	Evaluations' results and prescription

Test **Scenario**	The aim of this test scenario is to optimize the model developed by i4Q[DT] with the aim of decreasing the process time of the Festo plant. The model is going to be optimized through the variation of the number of parallel processing modules. First of all, a time range is going to be defined for each station. Then, each time variation is going to be simulated. Afterward, the simulations are going to be evaluated by means of the sum of the time that is required to build a mobile phone carcase in normal working condition. Lastly, the model that requires the minimum manufacturing time is going to be prescribed.

Step #	Step **Details**	**Results**		**Pass / Fail / Not executed**
		Expected	**Actual**	
1	Compare the number of manufactured products of the original configuration of the model with the prescribed model configuration	The prescribed model's performance is better than the original's	The prescribed model can manufacture 2 more products in 250s than the original	Pass

Table 20. Test Case i4QPA_TC-01

i4Q **Solution**	i4Q[PA] - Prescriptive Analysis Tool
Test Case **ID**	i4QPA_TC-02
Test Case **Type**	Performance
Test Case **Description**	The performance of the app is going to be tested by calculating the time required to do a prescription
Date Tested	2024/01/08
Module Stations **Involved**	All of them, as the model tries to simulate the general behaviour of the pipeline
Test Case **(Pass/Fail/Not Executed)**	Pass

Step #	**Prerequisites**		Step #	Test **Data**

1	Obtain Festo plant's model created by i4Q[DT]		1	N/A
2	Define model's parameters' range		2	N/A
3	Simulate model's variations		3	Simulations' results
4	Evaluate simulations results and prescribe minimum process time model		4	Evaluations' results and prescription

Test **Scenario**	The aim of this test scenario is to test the performance of the i4Q[PA] solution by carrying out a full prescription. The most time-consuming tasks of the solution are the simulation and the evaluation. First, a range of parallel processing modules is going to be defined for each station. Then, each time variation is going to be simulated. Afterward, the simulations are going to be evaluated by means of the sum of the time that is required to build a mobile phone carcase in normal working condition. Lastly, the model that requires the minimum manufacturing time is going to be prescribed.

Step #	Step **Details**	**Results**		**Pass / Fail / Not executed**
		Expected	**Actual**	
1	Measure the time it takes to simulate the defined range of model values	Unknown a priori	The solution is able to simulate 27 simulations of 300s with a step of 0.1 in 184.4s	Pass
2	Measure the time it takes to evaluate the simulations	Unknown a priori	The solution is able to evaluate 27 simulations of 300s with a step of 0.1 in 5s	Pass

Table 21. Test Case i4QPA_TC-02

3.11.2 Results and conclusions

The main result and conclusion of these tests is that the solution works properly and is easy to use by non-expert users.

3.12 Test Case of i4Q^{LRT} - Manufacturing Line Reconfiguration Toolkit

3.12.1 Test prerequisites, data and step details

In this section, the main tests carried out to validate the implementation of the i4Q Line Reconfiguration Toolkit solution in the Generic Pilot are presented.

i4Q **Solution**	i4Q^{LRT} - Manufacturing Line Reconfiguration Toolkit
Test Case **ID**	i4QLRT_TC-01
Test Case **Type**	Optimization
Test Case **Description**	Optimization of the configuration parameters of the pressure module of the Festo plant.
Date Tested	TBD
Module Stations **Involved**	CP Muscle Press
Test Case **(Pass/Fail/Not Executed)**	TBD

Step #	**Prerequisites**		Step #	Test **Data**
1	Measure and analyze the pressure values of the FESTO module.		1	N/A
2	To analyze the maximum and minimum at which the casing is correctly added to the phone without any errors.		2	Simulations' results
3	To prepare the model to analyze the data from the FESTO station to warn of a better reconfiguration of the parameters.		3	N/A

Test **Scenario**	The objective of this test scenario is to develop a model for i4Q^{LRT} to decrease the state of the "CP Muscle Press" pressure module. The model will be optimized by obtaining historical pressure data and results. The result, after the execution of the solution, is to provide reconfiguration parameters of the pressure module to obtain the best quality result.

Step #	Step **Details**	**Results**		**Pass / Fail / Not executed**
		Expected	**Actual**	
1	Launch the solution	The values are correct for the model	TBD	TBD

2	Launch the solution	The values are not correct for the model. Provide new reconfiguration values.	TBD	TBD

Table 22. Test Case i4Q LRT_TC-01

3.12.2 Results and conclusions

Results and conclusions will be established at the time of the test case execution.

4. Conclusions

Deliverable **D6.10 v3** extends and completes the technical information of the Module Stations described in the deliverable D6.7, providing detailed information, such as a Generic Pipeline, server configuration in terms of all the necessary software infrastructure, network characteristics and connection parameters, among the most important.

The availability and use of the Generic Pilot experimental facilities will allow performing experiments of the i4Q Solutions in a system beyond the environments defined for the Pilot, to facilitate the iterative improvement of i4Q Solutions. It is important to be aware that these facilities are focused on teaching and training environments and don´t have the complexity of being able to interact or obtain improvements similar to those of the industrial environment.

Test Cases consist of a series of very specific steps and under the consideration of very specific parameters that i4Q Solutions Providers execute to test their i4Q Solutions in certain scenarios. A standard procedure has been used to perform the Test Cases, that clearly and very synthetically defines the steps to be followed to perform the experiment or test.

The second part of D6.10 v3 is dedicated to collect the results of the Test Cases executed by each i4Q Solution, which in order to be able to interpret them correctly it is necessary to take into consideration that as it was said in the execute summary, that is to say, Module Stations are not a real industrial production environment such as those that exist in the industrial production environment in the i4Q Pilots' factories, and therefore their technical characteristics are limited, that there are certain difficulties in obtaining the information that i4Q Solutions Providers may require in order to perform their Test Cases with a very limited access to certain aspects of the i4Q Solutions, and finally, it will be necessary establish a schedule of execution of Test Cases and given the special characteristics of accessibility to the test environment.

References

FESTO (2022a). CP Factory High-bay storage for pallets CP-F-ASRS32-P. Available: https://ip.festo-didactic.com/InfoPortal/CPFactoryLab/hardware/station/details.php?model=CP-F-ASRS32-P

FESTO (2022b). CP Factory Branch (CP-F-BRACH). Available: https://ip.festo-didactic.com/InfoPortal/CPFactoryLab/hardware/base/details.php?lang=en&model=CP-F-BRANCH

FESTO (2022c). Robotino. Available: https://www.festo-didactic.com/es-es/servicio-y-asistencia/robotino/?fbid=ZXMuZXMuNTQ3LjE0LjM0LjExNjg

FESTO (2022d). CP Lab Branch (CP-L-BRANCH). Available: https://ip.festo-didactic.com/InfoPortal/CPFactoryLab/hardware/station/details.php?model=CP-L-BRANCH&lang=en

FESTO (2022e). Camera inspection (CP-AM-CAM). Available: https://ip.festo-didactic.com/infoportal/CPFactoryLab/hardware/application/details.php?model=CP-AM-CAM&lang=en

FESTO (2022f). iDrilling (CP-AM- iDRILL). Available: https://ip.festo-didactic.com/InfoPortal/CPFactoryLab/hardware/application/details.php?model=CP-AM-iDRILL&lang=en

FESTO (2022g). Magazine (CP-AM-MAG). Available: https://ip.festo-didactic.com/InfoPortal/CPFactoryLab/hardware/application/details.php?model=CP-AM-MAG&lang=en

FESTO (2022h). Muscle press (CP-AM-MPRESS). Available: https://ip.festo-didactic.com/InfoPortal/CPFactoryLab/hardware/application/details.php?model=CP-AM-MPRESS&lang=en

FESTO (2022i). Output (CP-AM-OUT). Available: https://ip.festo-didactic.com/InfoPortal/CPFactoryLab/hardware/application/details.php?model=CP-AM-OUT&lang=en

FESTO (2022j). Universal Robots (UR5). Available: https://www.festo.com/es/es/e/productos/landing-pages-id_326973/

i4Q. (2022). I4Q Solutions. Available: https://i4q.upv.es/