D1.9 – Requirements Analysis and Functional Specification v2

WP1 – Industrial Scenarios and Requirements Analysis

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	D1.9 – Requirements Analysis and Functional Specification v2		
WORK PACKAGE	WP1 – Industrial Scenarios and Requirements Analysis		
DATE OF DELIVERY	CONTRACTUAL: Sep 2021	ACTUAL	Sep 2021
NATURE	Report	DISSEMINATION LEVEL	Public
LEAD BENEFICIARY	TUB		
RESPONSIBLE AUTHOR	Anna M. Nowak-Meitinger (TUB), Jan Mayer (TUB), Roland Jochem (TUB)		
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, 19-WHI, 20-BIES, 21-FACT, 22-RIAS, 23-FARP, 24-FIDIA		
TARGET AUDIENCE	1) i4Q Project partners; 2) industrial community; 3) other H2020 funded projects; 4) scientific community		
DELIVERABLE CONTEXT/ DEPENDENCIES	D1.9 is the second deliverable of Task 1.4 Requirements Analysis and Functional Specification. • The preceding document was submitted in April 2021 and includes the first version of Requirements Analysis and Functional Specification (D1.4). It was focused on the first version of pilot-specific and solutions-related requirements, their elicitation, analysis, validation, and documentation. • D1.9 is focused on the second version of pilot-specific and solutions-related requirements and the definition of functional specifications. D1.9 has a strong relation to • D1.8 Demonstration Scenarios and Monitoring KPIs Definition v2 D1.9 provides input for and receives input from • D2.3 Report of Business Viewpoint • D2.4 Report on Usage Viewpoint • D2.5 Functional Specifications • D2.6 Technical Specifications • D2.7 Reference Architecture and Viewpoint Analysis		

	v2 D1.9 provides input for WP3, WP4, WP5 and WP6.
EXTERNAL ANNEXES/ SUPPORTING DOCUMENTS	None
READING NOTES	None
ABSTRACT	The deliverable D1.9 *Requirements Analysis and Functional Specification v2* is a Functional Specification document that provides an in-depth definition of the functionalities/behaviours of all i4Q Solutions and explains how related requirements will be fulfilled. It pursues the requirements engineering process that has been started in D1.4, by eliciting, discussing, and refining all pilot and solution requirements. Function Structure Diagrams (FSD) for all i4Q Solutions are created as a basis for the functional specification and the exact mapping of the requirements to the solutions. A suitable methodology and approach are developed and performed based on standards and guidelines such as ISO/IEC/IEEE 29148, ISO/IEC/IEEE 12207, and ISO/IEC/IEEE 15288, VDI 2221 and VDI 2206. In accordance to the method of Function Based Systems Engineering (FBSE), FSDs are used to identify and specify the (sub-) functions, the data flow, as well as the input and output of each i4Q Solution. Overall, the Functional Specification contains a brief description of the main functions and objectives of each solution, the FSD, including the processes and interfaces, the description of the to-be situation through the mapped requirements of users and the interoperability with other solutions and systems in the technical structure with its interfaces. To analyse the strengths and weaknesses of the solutions, the functional specifications are evaluated and explained in their completeness, precision, interface specifications, and requirements origins to define the further process in the BUILD work packages. Validation and verification of the requirements fulfilment will be elaborated when all solutions will be implemented and evaluated in the pilot factories (WP6).

Document History

VERSION	ISSUE DATE	STAGE	DESCRIPTION	CONTRIBUTOR
0.1	10-Feb-2021	ToC	Table of content and sections allocation to partners	TUB
0.2	08-June-2021	ToC	Table of content ready for review and sections allocation to partners	TUB
0.3	18-06-2021	ToC	Include feedback from partners in the ToC	TUB
0.4-0.21	02-09-2021	Draft	Contributions to Sections 1, 2, 3, 4, 5 Adjustments in ToC	TUB
0.22	03-09-2021	Draft	Review of pilot and solution subsections in Sections 3 and 4	TUB, pilots, solution providers
0.23	07-09-2021	Draft	Internal Review	CERTH, FACTOR
0.24-0.26	15-09-2021	Draft	Address comments from internal reviewers	TUB
1.0	16.09.2021	Final draft	Submission for final internal review	TUB
2.0	30-09-2021	Final doc	Quality check and issue of final document	CERTH

Disclaimer

Copyright message

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS/ACRONYMS

AI	Artificial Intelligence
API	Application Programming Interface
AV	SysML - ARCHITECTURE View
BP	Business Process
CNC	Computer Numerical Control
CMMS	Computerized Maintenance Management System
D1.4	i4Q Deliverable D1.4 - Requirements Analysis and Functional Specification
D1.8	i4Q Deliverable D1.8 - Demonstration Scenarios and Monitoring KPIs Definition v2
D1.9	i4Q Deliverable D1.9 - Requirements Analysis and Functional Specification v2
D2.3	i4Q Deliverable D2.3 - Report of Business Viewpoint
D2.4	i4Q Deliverable D2.4 - Report on Usage Viewpoint
D2.5	i4Q Deliverable D2.5 - Functional Specifications
D2.6	i4Q Deliverable D2.6 - Technical Specifications
D2.7	i4Q Deliverable D2.7 - Reference Architecture and Viewpoint Analysis v2
DB	Database
DDS	Data Distribution Service
ERP	Enterprise Resource Planning
FBSE	Function-Based Systems Engineering
FFBD	Functional Flow Box Diagrams
FSD	Function Structure Diagram
GDPR	General Data Protection Regulation
HSM	Hardware Security Module
IACS	Industrial Automation and Control Systems
ICT	Information and communications technology
ID	Identifier
IEC	International Electrotechnical Commission

IEEE	Institute of Electrical and Electronics Engineers
IIoT	Industrial Internet of Things
IoT	Internet of Things
ISO	International Organization for Standardization
KPI	Key Performance Indicator
LPWAN	Low-Power Wide-Area Network
MBSE	Model-Based Systems Engineering
MES	Manufacturing Execution System
ML	Machine Learning
MPFQ	Material-Process-Functions-Quality
MS	Milliseconds
NFR	Non-Functional Requirement
OEE	Overall Equipment Effectiveness
OPE	Overall Production Effectiveness
PC	Pilot Case
PLC	Programmable Logic Controller
PPM	Parts per Million
PVA	Polyvinyl Alcohol
QC	Quality Control
REQ	Requirement
REST	Representational State Transfer
RIDS	Reliable Industrial Data Services
SAV	SysML - System ANALYSIS View
SDN	Software Defined Networks
SDV	SysML - System DESIGN View
SE	Systems Engineering
SIV	SysML - System IMPLEMENTATION View
SITV	SysML - System INTEGRATION & TEST View
SOA	Service Oriented Architecture
SP	Solution Providers

SRV	SysML - System REQUIREMENTS View
TC	Test Case
TCM	Tool Condition Monitoring
TCP	Transmission Control Protocol
TSCH MAC	Time Slotted Channel Hopping and Media Access Control
TSN	Time-Sensitive Networking
URLLC	Ultra-Reliable and Low Latency Communication
USB	Universal Serial Bus
VDI	The Association of German Engineers
WP	Work Package
WPAN	Wireless Personal Area Network
WSN	Wireless Sensor Network
ZDM	Zero-Defects Manufacturing

Executive Summary

Deliverable D1.9 *Requirements Analysis and Functional Specification v2* provides the complete sets of user requirements for the business processes of the six pilot cases and their mapping to the 22 i4Q Solutions as well as the Functional Specification of the Reliable Industrial Data Services (RIDS). In the first deliverable of this task (D1.4), the basis has been formed when through technical discussions between end-users and technical providers, the gaps between the AS-IS and TO-BE scenarios have been analysed and the requirements elicited. Deliverable D1.9 provides a deeper understanding and analysis of the pilot requirements and describes by which solution functions these requirements will be fulfilled. Also, the interoperability of the i4Q Solutions is considered by defining functional interfaces and eliciting the corresponding requirements. This is an important input for the DESIGN work package 2, in which the business, usage, functional and implementation viewpoints are elaborated in detail. In D1.8 *Demonstration Scenarios and Monitoring KPIs Definition v2*, associated KPIs of the pilot use case scenarios are developed and related to the requirements.

A suitable methodology and approach are developed and performed based on standards and guidelines such as ISO/IEC/IEEE 29148, ISO/IEC/IEEE 12207 and ISO/IEC/IEEE 15288, VDI 2221 and VDI 2206, and in accordance to the method of Function-Based Systems Engineering (FBSE) (INCOSE 2017, p. 292ff). The open-source systems modelling language SysML[1] is used for creating diagrams of the user requirements and the mapping of the requirements to the i4Q Solutions. To initiate defining a functional specification of the i4Q RIDS, Function Structure Diagrams (FSD) (Bender and Gericke 2021, p. 241f) are developed and used to identify and specify the (sub-)functions, the data flow, as well as the input and output of each i4Q Solution. The FSD serve as a basis for technical discussions and the mapping of requirements to functions to specify the functionality of each solution.

Several evaluation criteria, like precision, interfaces and completeness are introduced to assess the technical i4Q Solutions according to the mapping of the requirements. This enables a coherent status overview regarding the challenges of each i4Q Solution in relation to the customers' needs.

The main results of this document are:

- The overview of the complete sets of requirements for the business processes of each pilot and the mapping of these requirements to the functions of the i4Q Solutions.
- Function Structure Diagrams (FSD) including the (sub-)functions, the data flow, as well as the input and output of each i4Q Solution and their interfaces to other i4Q Solutions.
- Complete sets of requirements for each i4Q Solution and mapping to the functional architecture, derived from the FSD.

[1] www.sysml.org

- The evaluation of the functional specification of all i4Q Solutions including the requirements analysis results and the explanation of the further use of these functional specifications.
- Outlook of further requirements engineering in the BUILD and EVALUATE phases of the i4Q Project.

Document Structure

Section 1: The *Introduction* describes the content of this deliverable in terms of its evolutionary development, the context to other deliverables, and the impact to upcoming work packages.

Section 2: In *Methodology and Approach*, the initial situation with the two perspectives in i4Q Project is described first. Then the methods of requirements engineering and systems modelling including the ISO standards, VDI guidelines, Model-Based Systems Engineering (MBSE) and the system modelling language SysML are explained. Finally, the procedure for requirements analysis and functional specification, including the four iteration steps as well as diagram modelling, mapping of requirements to solutions and evaluation of the same, is explained. The final result is the evaluation of the Functional Specification.

Section 3: The *Requirements* are clustered in *Pilot Requirements* and *Solution Requirements*. All requirements are described in SysML format with their hierarchy structures and mappings to i4Q Solution functions.

Section 4: The *Functional Specification* for all 22 i4Q Solutions is described via Function Structure Diagrams (FSD) and the mapped requirements. An analysis with an evaluation of the results is also included. In addition, tables with unmapped and general requirements are provided.

Section 5: The *Evaluation and Use of Results* is described, considering the methodological approach, the analysis and specification results of the i4Q RIDS and their overall context. Also, the use of these results in the upcoming work packages is discussed.

Section 6: The *Conclusions* give an overview of the main findings and improvements achieved in this deliverable and how they were accomplished, considering the whole Task 1.4. Additionally, a short outlook on the upcoming work packages is given.

1. Introduction

1.1 Evolution of Requirements Analysis and Functional Specification

This deliverable D1.9 is the second and final deliverable of Task 1.4 *Requirements Analysis and Functional Specification*. The preceding deliverable D1.4 was submitted in April 2021 and is focused on the elicitation, analysis, validation, and documentation of the first version of pilot-specific and solutions-related requirements. The requirements elicitation was mainly based on technical discussions between end-users and technical providers, in which the gaps between the AS-IS and TO-BE scenarios have been analysed. Templates were used for the elicitation and documentation of requirements. Also, generic descriptions of the i4Q Solution functions were included as well as the mappings of both.

D1.9 focuses on the second version of pilot-specific and solutions-related requirements which provide a deeper understanding and clearer picture of the pilots and their needs as well as of the functions of all i4Q Solutions. The modelling language SysML[2] supports the documentation of the complex structures of requirements as well as the functional structures of the i4Q Solutions. A precise mapping of requirements to i4Q Solutions is provided as well as the definition of functional specifications through the requirements and Function Structure Diagrams (FSD). Finally, an analysis and discussion of the results is performed and documented in this deliverable and an outlook for further work regarding the requirements analysis and functional specification is given.

1.2 Context to other Deliverables

D1.9 has a strong relation to deliverable D1.8 *Demonstration Scenarios and Monitoring KPIs Definition v2* in which the KPIs for each business process of the pilots are described. In both deliverables, pilot-specific needs and processes are described and evaluated from two different perspectives. Figure 1 outlines the connection between both tasks and the appropriate deliverables.

Furthermore, D1.9 provides input for and receives input from the viewpoint tasks in work package 2. The deliverables D2.3 *Report of Business Viewpoint*, D2.4 *Report on Usage Viewpoint*, D2.5 *Functional Specifications*, D2.6 *Technical Specifications* are related to the requirements and functional specifications provided by deliverable D1.9. The work on all these deliverables runs in parallel since they are submitted at the same time. This requires a clarification of interfaces, connections and input and output in-between the deliverables.

[2] www.sysml.org

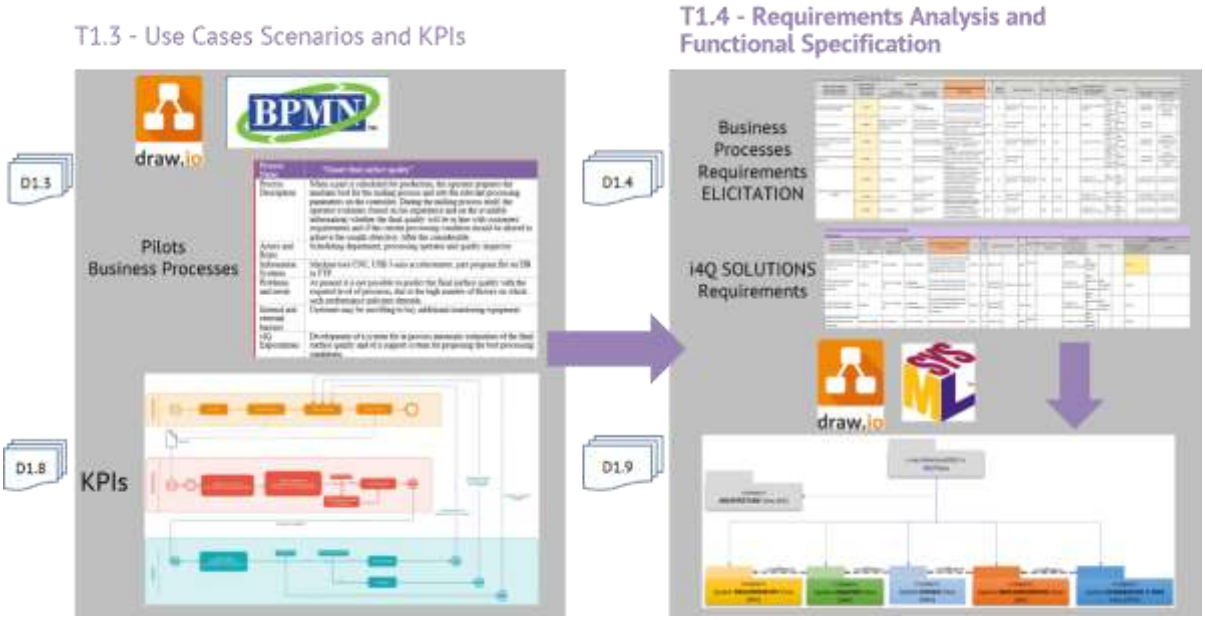


Figure 1. The Connection between Task 1.3 and Task 1.4 in i4Q Project

Deliverable D2.5 *Functional Specifications* provides a deeper analysis of the i4Q Solution functions as well as their inputs and outputs and takes the FSD and related requirements of D1.9 as an input (Figure 2). Furthermore, D2.7 *Reference Architecture and Viewpoint Analysis v2* uses sub-functions for the mapping to the RA and provides descriptions of each i4Q Solution.

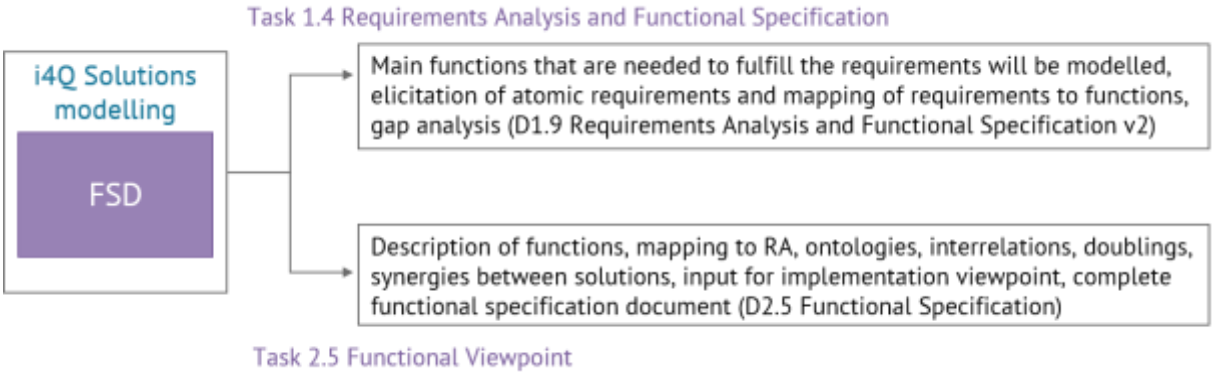


Figure 2. The Connection between Task 1.4 and Task 2.5 in i4Q Project

1.3 Impacts and Benefits for Work packages 3 to 6

D1.9 provides input for the upcoming BUILD work packages (WP3, WP4, WP5) in which the i4Q Solutions will be built. The solution-specific requirements and the FSD, which together form the Functional Specification in this deliverable, might be a roadmap for the solution providers to keep track of the scope of the i4Q Solutions and specify the functions. Also, for the EVALUATE work package (WP6), deliverable D1.9 provides the basis in the form of pilot requirements and the i4Q Solutions mapped to the pilot business processes. The validation and verification of i4Q Solutions is performed during the requirements review.

2. Methodology and Approach

2.1 Initial Situation

The initial situation for the requirements engineering in i4Q Project with its six Pilots and 22 planned and predefined i4Q Solutions is described in the preceding deliverable D1.4. To recapitulate this, **Error! Reference source not found.** shows the two perspectives which have to be considered in the requirements elicitation process.

Figure 3. The two Perspectives of Requirements for the i4Q RIDS

The process of defining and connecting requirements and functional specifications in i4Q Project is outlined in Figure 4. This process is supported by an approach that allows partners to contribute and document their requirements and functional specifications in a traceable and comparable way. The standards and methods used to form the basis for the work in this requirements analysis are described in Section 2.2. The specific procedure that led to the results in this deliverable is explained in Section 2.3.

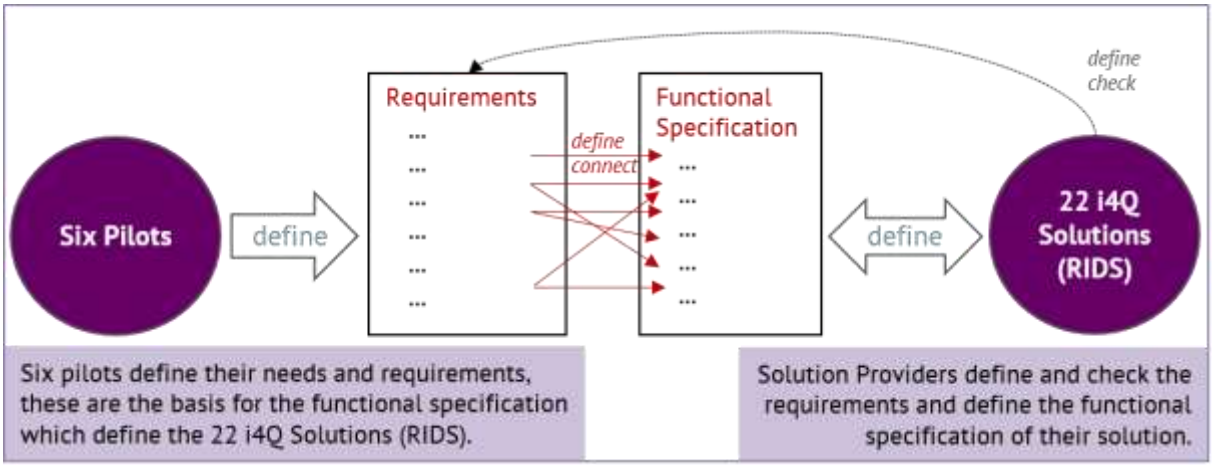


Figure 4. Process of Requirements Analysis and Functional Specification with Involved Partners

2.2 Method Framework

2.2.1 Standards and Guidelines used in i4Q Project

The standards and guidelines that are used for requirements elicitation and analysis in i4Q Project are described in the preceding deliverable D1.4 and listed below:

- ISO/IEC/IEEE 29148 (ISO/IEC/IEEE 29148:2018-11)
- ISO/IEC/IEEE 12207 (ISO/IEC/IEEE 12207:2017-11)
- ISO/IEC/IEEE 15288 (ISO/IEC/IEEE 15288:2015-05-15)
- VDI 2221 (VDI 2221-1:2019-11 and VDI 2221-2:2019-11)
- VDI 2206 (VDI 2206:2004-06)

These standards and guidelines define the basic procedure of the "Stakeholder Needs and Requirements Definition process" (ISO/IEC/IEEE 12207:2017-11, p. 59ff) as well as the design of technical products and systems in VDI 2221 and a design methodology for mechatronic systems in VDI 2206 including the aspects of requirements engineering.

In i4Q Project, these standards and guidelines support requirements engineering in terms of completeness and strategy of the process to achieve complete sets of requirements, considering the initial situation described above. This is important to define the basis for ensuring an appropriate design of the i4Q Solutions, as the requirements form the basis for the system design and the assurance of properties during system integration when the i4Q Solutions are verified and validated against the defined requirements. Figure 5 shows the V-model defined in VDI 2206:2004 as a macro-cycle with the assigned work packages of the i4Q project and how the work flow is planned. In this process, the requirements go into the left wing of the V-model as the starting point for the system design. This correlates with Task 1.4, but the system design process is in this deliverable also already started with the description of a functional specification. In the later version of VDI 2206 presented in Section 5.3, the approach of the requirements engineering is different.

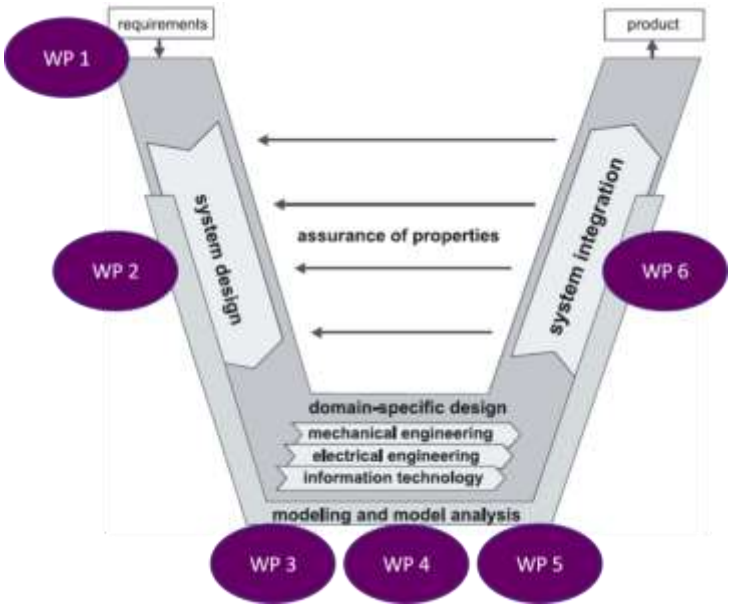


Figure 5. V-Model as a macro-cycle (VDI 2206:2004-06, p. 29) with assigned work packages of the i4Q Project

2.2.2 Systems Modelling with SysML

Systems Engineering is an interdisciplinary and holistic approach for the definition, documentation, verification, and validation of system requirements with respect to all aspects of the lifecycle of the system (Weilkiens 2014, p. 11; INCOSE 2017, p. 15f). Model-Based Systems Engineering (MBSE[3]) is the formalized application of systems modelling to support, but not limited to, system requirements activities (Weilkiens 2014, p. 21). MBSE increases the ability to capture, analyse, communicate, and manage information associated with the specification of a product, resulting in improved communication, control of system complexity, product quality, and knowledge capture (INCOSE 2017, p. 291). A system model should always be centrally available (developers always have access to the latest version), formal (unambiguous, computer interpretable), and complete, coherent, and consistent, so that the relationship information between elements is explicitly included (Zafirov 2014, p. 82).

For the complex task of requirements engineering in the i4Q project, the system modelling language SysML[4] is used to model the requirements and functions and to achieve a comprehensive understanding of the *system* or the *system of systems* that describes it independently of any discipline. The communication between different disciplines and their teamwork shall be specifically supported by the modelling of the various aspects of the i4Q Solutions. In this way, errors are to be avoided and product quality is to be increased. In addition, the product development process can be shortened. The application of systems engineering is always individually tailored to the use case. Through systems modelling, developers should be able to communicate and exchange information about a system or product with its problems, functions and solutions from different perspectives, e.g., by using visualized diagrams (INCOSE 2017, p. 291).

Since there are 22 i4Q Solutions to be developed related to the six Pilots and there are 24 interdisciplinary partners involved, the objectives of using SysML as a modelling language in the requirements analysis and functional specification process is to provide a communication basis, reduce complexity and ensure that requirements will be met by the i4Q Solutions. To handle the huge amount of information and gather the complex requirements and solution structures, the diagrams are modelled in some standardized SysML shapes that are shown in Figure 6. This modelling supports the elicitation of complete, consistent, feasible, traceable, and validatable sets of requirements. Tables are not able to structure, link, analyse and validate all these requirements in a comprehensible way. The functional specification refers to the requirements, and can be visualized in the models following the Function-Based Systems Engineering (FBSE), in which the functions of the system are decomposed iteratively into sub-functions and the

[3] https://mbseworks.com
[4] www.sysml.org

performance requirements are mapped (INCOSE 2017, p. 292ff). In this way, synergies, conflicts, which is helpful for the system design performed in WP 2.

The software tools diagrams.net[5] (former draw.io) and Cameo System Modeler[6] are used for the diagram modelling. The shapes and models have been chosen for the project specific tasks and conditions. The exact procedure is explained in Section 2.3.

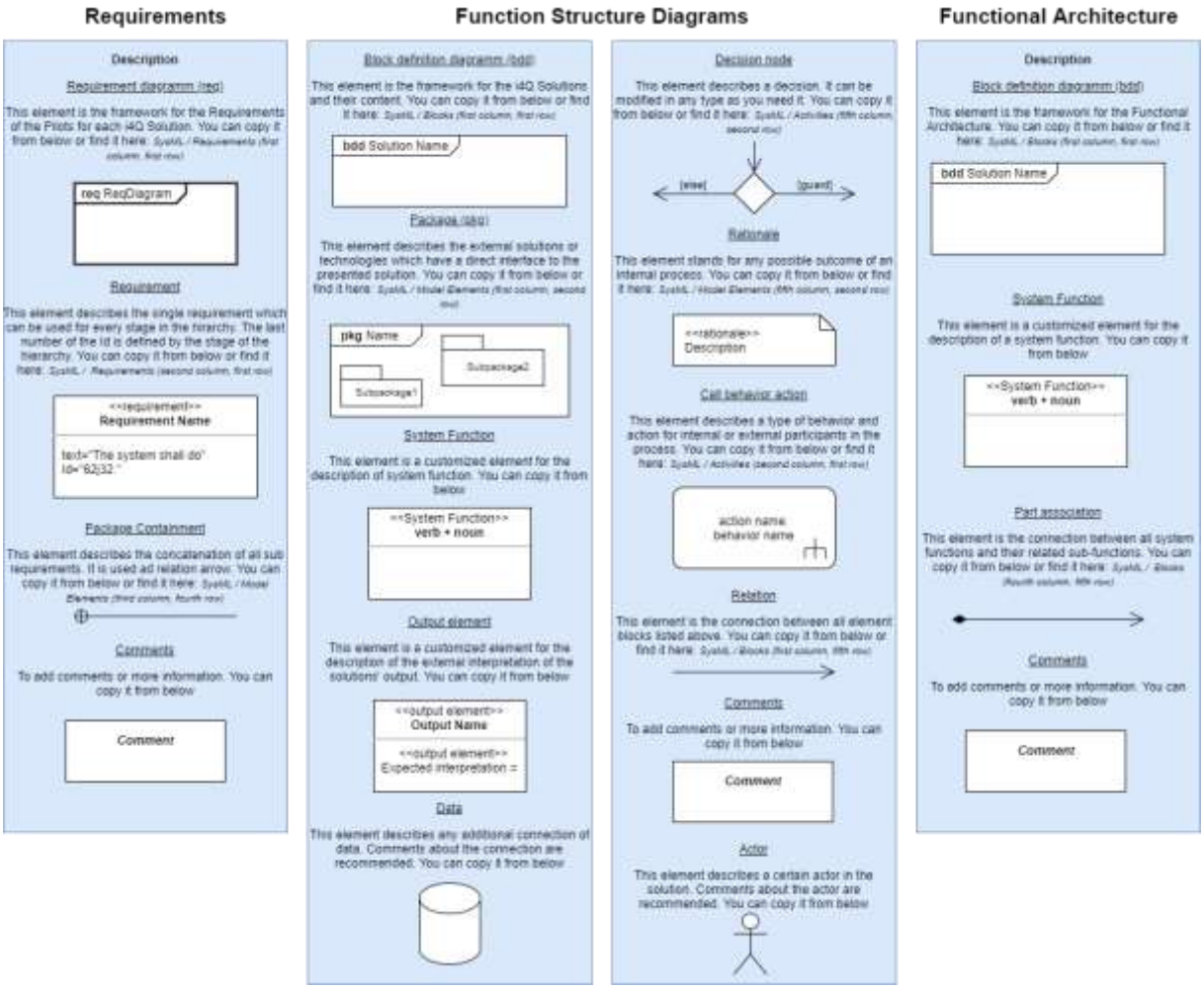


Figure 6. SysML Shapes used for Systems Modelling in i4Q Project

2.3 Procedure

2.3.1 Four Iteration Steps

To accomplish the requirements analysis and functional specification in this task, four iteration steps are defined and executed (Figure 7). The starting point is the first version of the deliverable D1.4, in which many requirements and functionalities have been documented at the

[5] https://app.diagrams.net/
[6] https://www.3ds.com/products-services/catia/products/no-magic/cameo-systems-modeler/

beginning of the project. These requirements and functions are iteratively reviewed and refined. The main purpose of the following four iteration steps is to gain a deeper knowledge of the 17 technical i4Q Solutions and the six pilot cases in which they are tested. The remaining 5 i4Q Solutions are guidelines and audit procedures which will be in use once the technical i4Q Solutions are established. Both the technical i4Q Solutions and pilot cases an interdisciplinary team and many views need to be brought to a common understanding of the situation. To handle this complex requirement and solution structure, the four iteration steps are defined as follows:

- The **first iteration step** is divided into two parts: the basic FSD of each functional solution (no guidelines) are modelled by the Solution Providers with the main functions taken from D1.4; and the basic pilot requirements are transformed from the lists in D1.4 into SysML diagrams.
- The **second iteration step** is also divided into two parts: the basic FSD are refined with detailed sub-functions and if applicable, with missing main functions. The Pilot requirement diagrams are refined and completed with specific requirements for the i4Q Solutions.
- In the **third iteration step,** the mapping of requirements to functions is realized in technical meetings with discussions about all the requirements. Also, the solution providers write requirements into the *Solution Requirement Template* to describe the interfaces between the solutions which are visible in the Inputs and Outputs of the FSD.
- In the **fourth iteration step,** the functional specification is derived by combining the FSD and mapped requirements for each solution. This involves identifying the specific functions of the solutions that will satisfy each requirement.

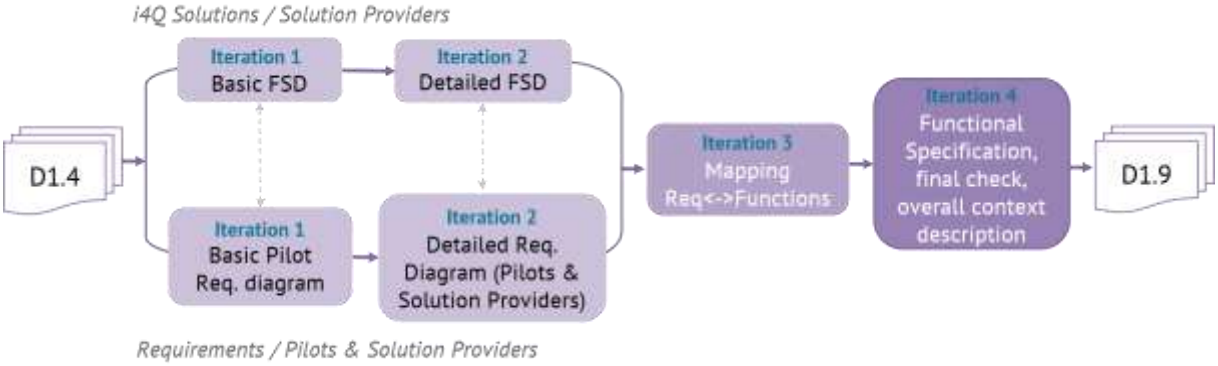


Figure 7. Four Iteration Steps in Document Evolution

During all iteration steps, there are interactions with the viewpoint tasks 2.3 to 2.6 and task 1.3 in order to model and refine the diagrams. Teamwork is a critical factor for success in this phase of the project, as the synchronization of all diagrams and their contents requires a well-organised collaboration of all partners involved. The Solution Providers take the central role here, as they work with all pilots and with all other Solution Providers (Figure 8).

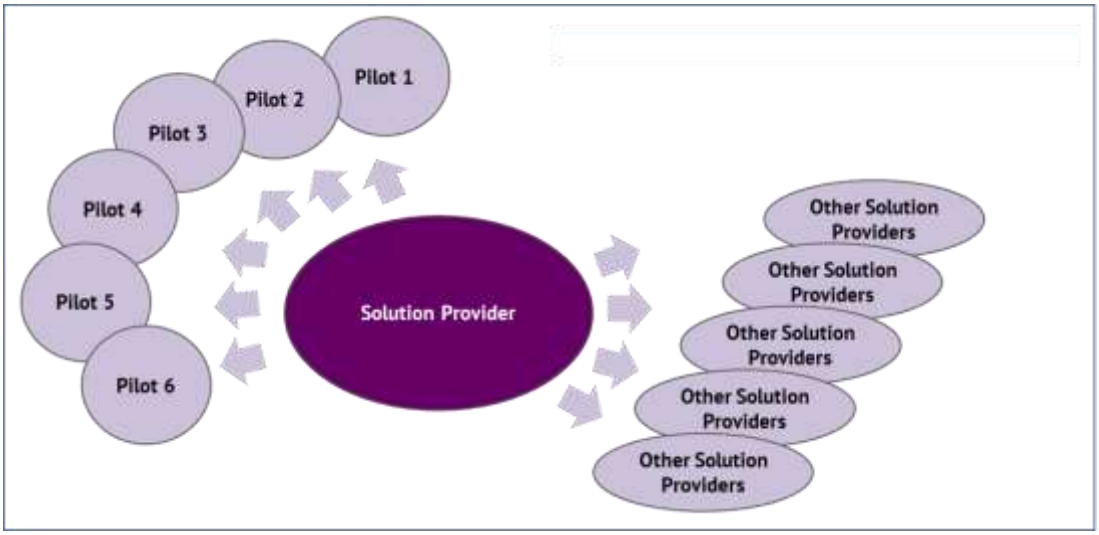


Figure 8. Central Role of Solution Providers in Requirements Analysis and Functional Specification

The four iteration steps are described in detail in the next sections including schematic diagrams of the pilot requirements and functions.

2.3.2 Pilot Requirements Diagram

The *Pilot Requirement Diagrams* are created by the pilots and their technical supporters based on the first interim result of requirement lists submitted in deliverable D1.4. The pilot requirement diagrams are structured in a tree structure for each business process from the top-level requirements to more specific requirements on the lower levels. These *application requirements* are refined and precisely defined in technical discussions.

Figure 9 outlines a schematic pilot requirement diagram. The requirements on the lowest level are called *atomic requirements*. The complete SysML diagrams are modelled with the open-source tool diagrams.net (former draw.io) and Cameo System Modeler. The results are presented in Section 3.

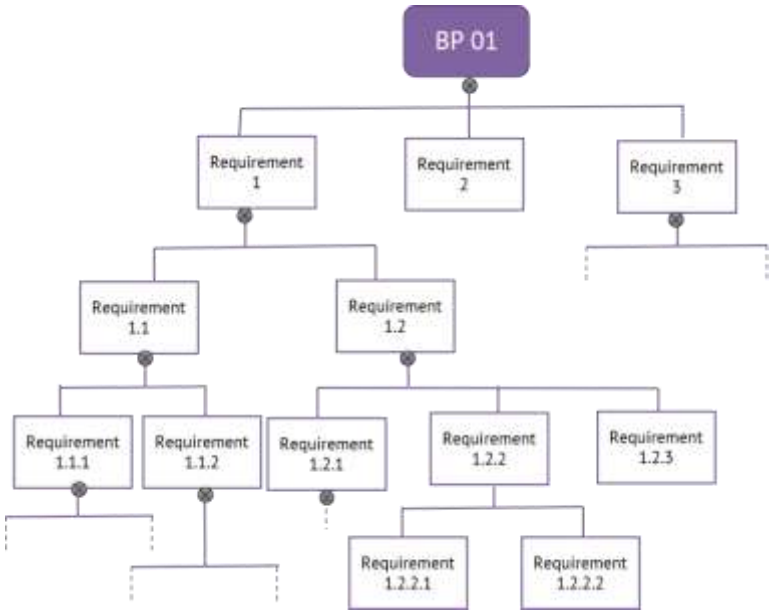


Figure 9. Schematic Pilot Requirements Diagram

2.3.3 Function Structure Diagram (FSD)

The Function Structure Diagram (FSD) is a tool to identify essential problems of the product, to describe essential functions, to show possible outlines of the product, to be able to discuss possible module divisions for a modular product, and to be able to map the functional description of the product architecture (Bender and Gericke 2021, p. 241f). Therefore, the main functions are described in function blocks and hierarchically decomposed into sub-functions, which form the functional architecture. These (sub-)function blocks are connected with arrows to define the data and information flow within the solution including their input and output (according to Bender and Gericke 2021, p. 243f). This procedure is similar to the Functional Flow Box Diagrams (FFBD) in the FBSE in which performance requirements are mapped to the functional components of the system (INCOSE 2017, p. 295ff). The system modelling language SysML can be used to describe the system model, in this case the FSD and the architecture (Zafirov 2014, p. 87).

The Function Structure Diagrams (FSD) of the i4Q Solutions are modelled based on the solution functions that have been roughly described in deliverable D1.4. In the first iteration, the FSD of i4Q Solutions include all main system functions as well as the input and output descriptions. In the second iteration, the main functions are reviewed and refined in sub-functions. Also, the input and output are updated considering the FSD of other i4Q Solutions. Different colours are used in the FSD to describe input, output and functional elements and make it easily understandable (Figure 10). A schematic FSD is outlined in Figure 11 with all (sub-)functions. The complete FSD of each i4Q Solution are presented in Section 4. All the requirements are mapped to the functional architecture in Section 4 (see Section 2.3.5).

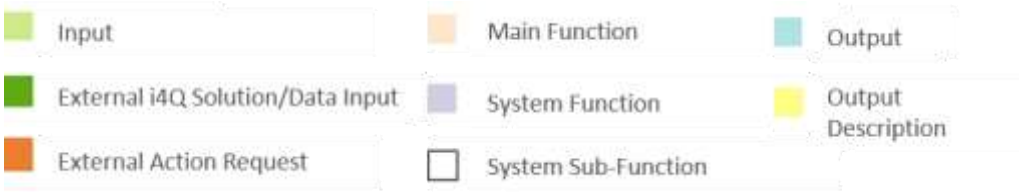


Figure 10. Colours used in the FSD to describe input, output and functional elements

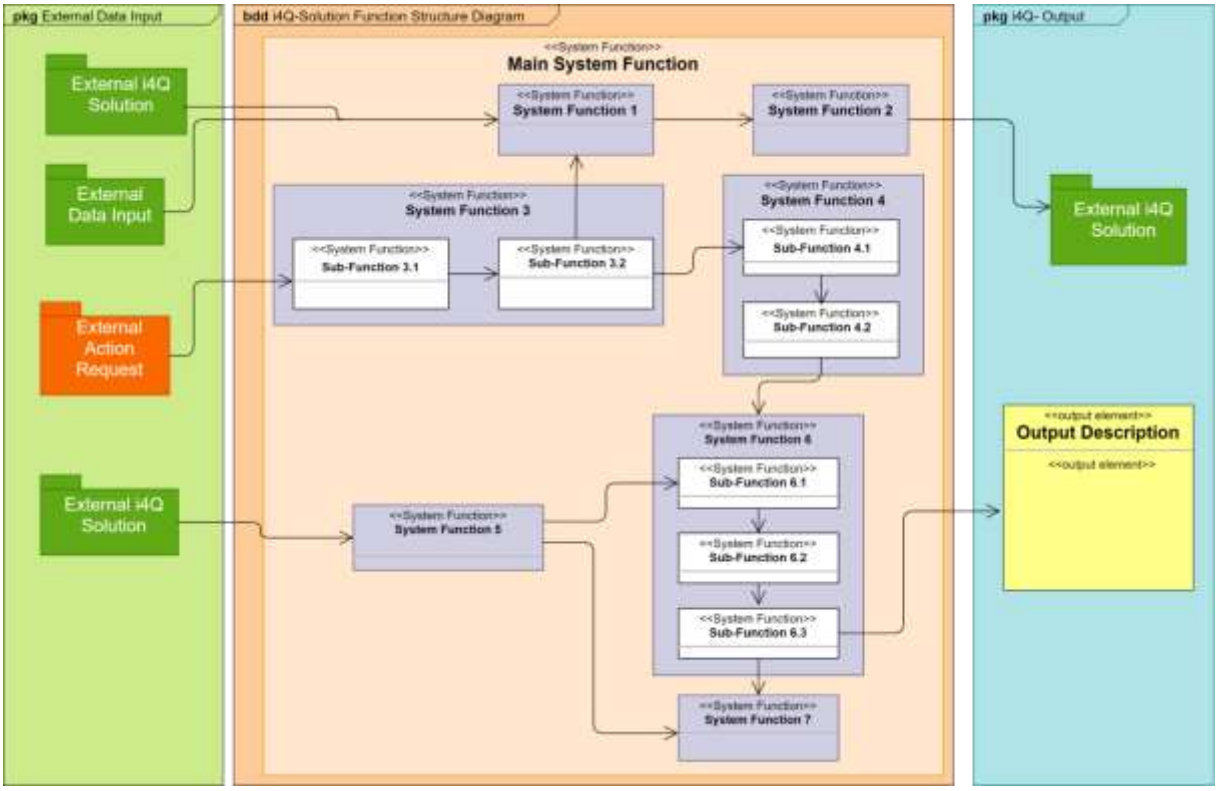


Figure 11. Schematic Function Structure Diagram (FSD)

2.3.4 Solution Requirement Template

To describe the solution specific requirements, which mainly result from the interfaces between the i4Q Solutions, a table named *Solution Requirement Template* is created, which is based on the *further requirement template* of D1.4. These interfaces are reflected in the connected solutions of the input and output boxes of the FSD. The full table of *interface requirements* is imported into Cameo System Modeler to be converted to SysML format and mapped to the FSD. The results are included in Section 4.

2.3.5 Mapping of Requirements to Functions

Through the mapping of the requirements to the functions and vice versa, the requirements are analysed and the solution functions are specified. The goal is to close all gaps and achieve a complete mapping of the lowest-level (sub-)functions to the atomic requirements of the business processes so that all requirements can be satisfied by the i4Q Solution.

The mapping is performed by the software tool Cameo System Modeler[7] and it is schematically shown in Figure 12. The results and complete mapping are presented in the Sections 3 and 4.

[7] https://www.3ds.com/products-services/catia/products/no-magic/cameo-systems-modeler/

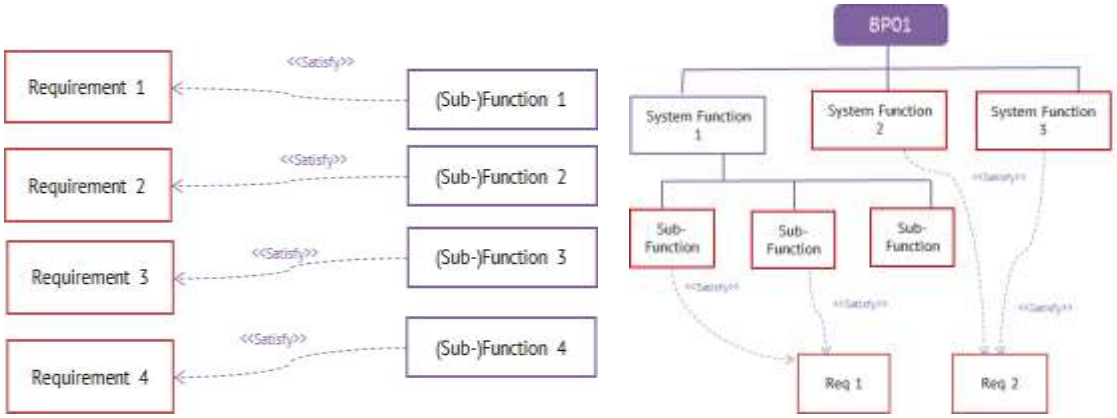


Figure 12. Schematic mapping of Requirements and Functions: decomposition diagram for the requirements (left) and functional architecture (right)

2.3.6 Functional Specification

The Functional Specification of the i4Q Solutions that is provided in this deliverable D1.9 consists of the FSD and the mapped performance requirements that specify the functions. This is realized by using the Cameo System Modeler software to create a valid data basis and to derive diagrams (see Section 2.3.5). Further, detailed descriptions of the system elements are provided in deliverable *D2.5 Functional Specification*.

2.3.7 Evaluation of Results

The 6-3-5 method was used to methodically develop suitable evaluation criteria for assessing the results of the requirements analysis and functional specification. These can also be found in a similar form in the INCOSE handbook regarding the Function-Based Systems Engineering, FBSE (INCOSE 2017, p. 296). The main aspects that are analysed using the evaluation criteria are defined and described in Table 1. There are four evaluation criteria that are applied to each solution in Section 4. The set of i4Q RIDS, i.e., all i4Q Solutions, are evaluated in Section 5 in the overall evaluation using the associated criteria listed in Table 1.

The results of the mapping and the resulting functional specification for each i4Q Solution are analysed and discussed in Section 4. The final results of the complete set of i4Q RIDS and the overall context description and evaluation is discussed in Section 5. The most important aspects and critical points are finally outlined as input for the following work packages 3 to 6.

Application to	Evaluation Criteria	Definition
Each i4Q Solution	*Completeness*	Ratio of (sub-)functions with assigned (mapped) requirements to functions without assigned requirements. This identifies "blank spots" in the diagrams where sub-functions do not meet any known requirement. This could also carry the risk that (as yet) unknown requirements are not met, while at the same time giving developers room to act. Furthermore, "hot spots" are identified, which on the one hand show which functions already have to fulfil clear requirements and on the other hand show which priorities have been set in the previous requirements engineering.
	Precision	Ratio of requirements mapped to sub-functions at the lowest level of the architecture to all requirements. The inverse is the ratio of requirements mapped to top-level functions to the total number of requirements. This is used to identify where further specification by users (pilots) and solution providers is required.
	Interface specification	Interfaces to other i4Q Solutions are examined and counted to identify dependencies and necessary specifications in the development of the i4Q Solutions to ensure interoperability. The definition and specification of the solution interfaces and their requirements mapped to the (sub)functions at the lowest level of the functional architecture are considered. From this, it can be derived where further interface specifications are required by the solution providers.
	Req Origin	Distribution of the origin of the requirements for a solution. From this, a possible diversity and expectation of the requirements originators can be derived. In particular, developers should know whose requirements they have to fulfil and who their stakeholders are.
Complete set of i4Q RIDS	*Req Origin comparison*	Diagram showing the number of Requirements (Req) and their Origin of all i4Q Solutions.
	Status comparison	Map showing the combination of Completeness, number of interfaces, and number of requirements.
	Interface comparison with solution map	Number of interfaces at the input and output side of the solutions. This shows whether a solution is more of a provider or a receiver. This is also made clear in the solution map.

Table 1. Evaluation Criteria for the Results of the Requirements Analysis and Functional Specification

To further analyse the overall status considering the evaluation criteria from Table 1, all technical i4Q Solutions are classified in four different areas of a coordination system which helps to understand further behavior and declare recommendations for single i4Q Solutions.

Figure 13 shows an example of the classification outcome. The metrics which are used for this classificatoin process are *Precision* on the y-axis, the normalized number of *Interfaces* on the x-axis and the total number of *Requirements* addressed to the i4Q Solutions as bubble size. The consumptions of the fields are as follows:

- **Left-upper field (A):** The mapping of the requirements to the i4Q Solutions is quite precise. Additionally, the number of interfaces in relation to other i4Q Solutions is low. This concludes in a detailed development description from Pilots and other stakeholders without having too much interaction with other i4Q Solutions from a requirements perspective.
- **Right-upper field (B):** The mapping of the requirements to the i4Q Solutions is quite precise. On the contrary, the number of interfaces is classified as high. This concludes in a detailed development description from pilots and other stakeholders, but simultaneously has a higher organizational effort for the interface definition and clarification.
- **Left-lower field (C):** The mapping of the requirements to the i4Q Solutions is not as clear as in other i4Q Solutions. At the same time, the number of interfaces is also low in the projects' aspects. This concludes in a generic development description from stakeholders without having too much interaction with other i4Q Solutions from a requirements perspective. As result, this provides freedom in software development by reaching overarching goals but also leaves some low-level functionalities unspecified by requirements.
- **Right-lower field (D):** The mapping of the requirements to the i4Q Solutions is not as clear as in other i4Q Solutions. Furthermore, the number of interfaces is quite high among all other i4Q Solutions. This concludes in a generic development description from stakeholders and having a higher interaction with other i4Q Solutions from a requirements perspective. Being part of this field results in a higher planning effort and risk for the positive outcome of the project, speaking about the needs which were addressed to this solution.

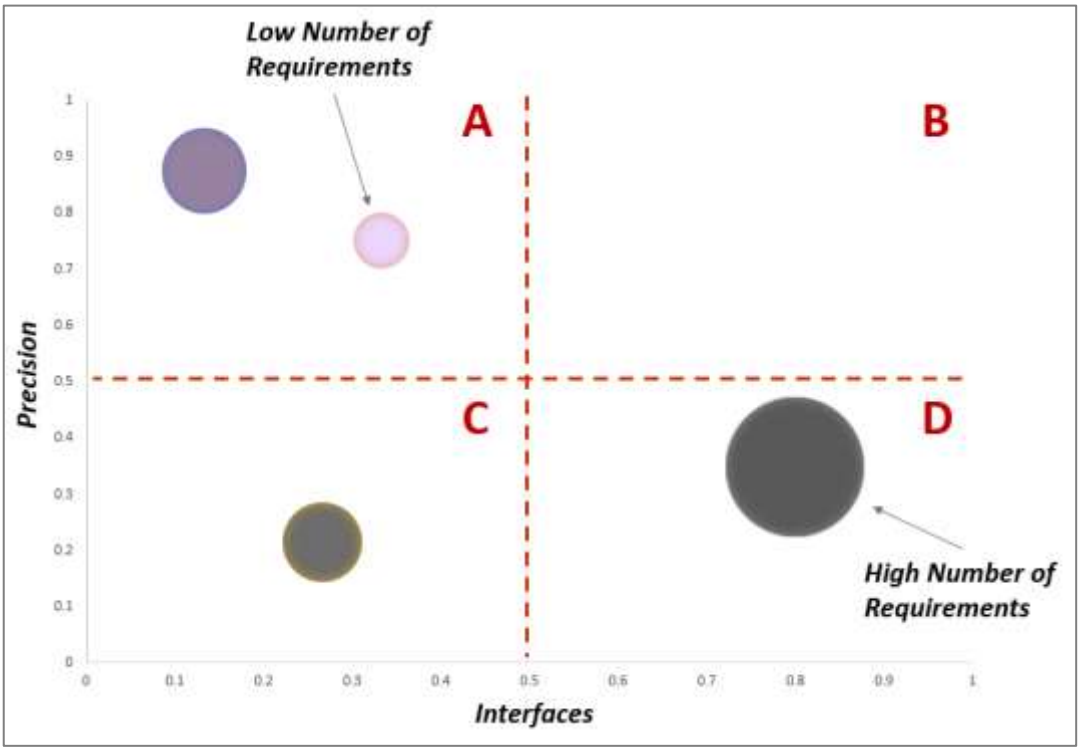


Figure 13. Schematic Status Comparison

3. Requirements Analysis Results

3.1 Pilot Requirements

This chapter presents the identified pilot requirements that relate to the IoT-based Reliable Industrial Data Services (RIDS) and thus to the 22 i4Q Solutions. The first version of the pilot requirements is included in D1.4 in the form of lists. Building on these lists, this section develops a deeper description and understanding of the requirements and their hierarchical structures using SysML diagrams. It also provides the precise mapping to the functions of the i4Q Solutions that satisfy the requirements.

The structure of the diagrams is described in Section 2.3.2, the mapping is explained in Section 2.3.5. These diagrams can be useful to verify and validate at the end of the project if all requirements could be fulfilled.

Since the mentioned pilot requirements are all performance requirements and are mainly of functional/technical requirement type, they are all of high priority and should be met by the i4Q Solutions. In addition, the type of non-functional requirements is optionally used to assign further information to the requirements.

3.1.1 Pilot 1: Smart Quality in CNC Machining

FIDIA's main objectives which needs to be addressed by i4Q Solutions are fully connected to the monitoring and adaption of processing conditions to guarantee the quality of workpieces is in line with customers' requirements. As main challenges, the pilot describes the need to collect and synchronize their data correctly to further ensure a safe data pipeline. With the established robustness of the background structure multiple smart decision support systems are required to predict the quality of the final surface of the workpiece, to detect chatter from the working machine and to evaluate the machine tool conditions.

Requirements Diagrams and Mapping of (Sub-)functions to Atomic Requirements

The Pilot Requirement Diagrams of the three business processes of Pilot 1 are shown in Figure 14, Figure 15 and Figure 16. Therein, the mapped i4Q Solutions are named. The diagrams show in white boxes all requirements that apply to all three business processes. The green boxes contain requirements for the specific business process. There are four main parent requirements based on the phases of the data flow management and utilization: Data Collection, Management, Analysis and new feature implementation. This fourth set of requirements changes in the three business processes taken into consideration: to predict surface roughness, to remove chatter and to identify failing components. Each parent requirement is further divided in specific requirements.

Figure 17 shows the requirements structure of the business processes of Pilot 1 with the assigned (sub-)functions of the mapped i4Q Solutions. Only the atomic requirements are mapped, since they are the most precise requirements on which the high-level requirements depend. The fulfilment of the requirements by the mapped solution functions can be measured in further steps (see Section 5.3).

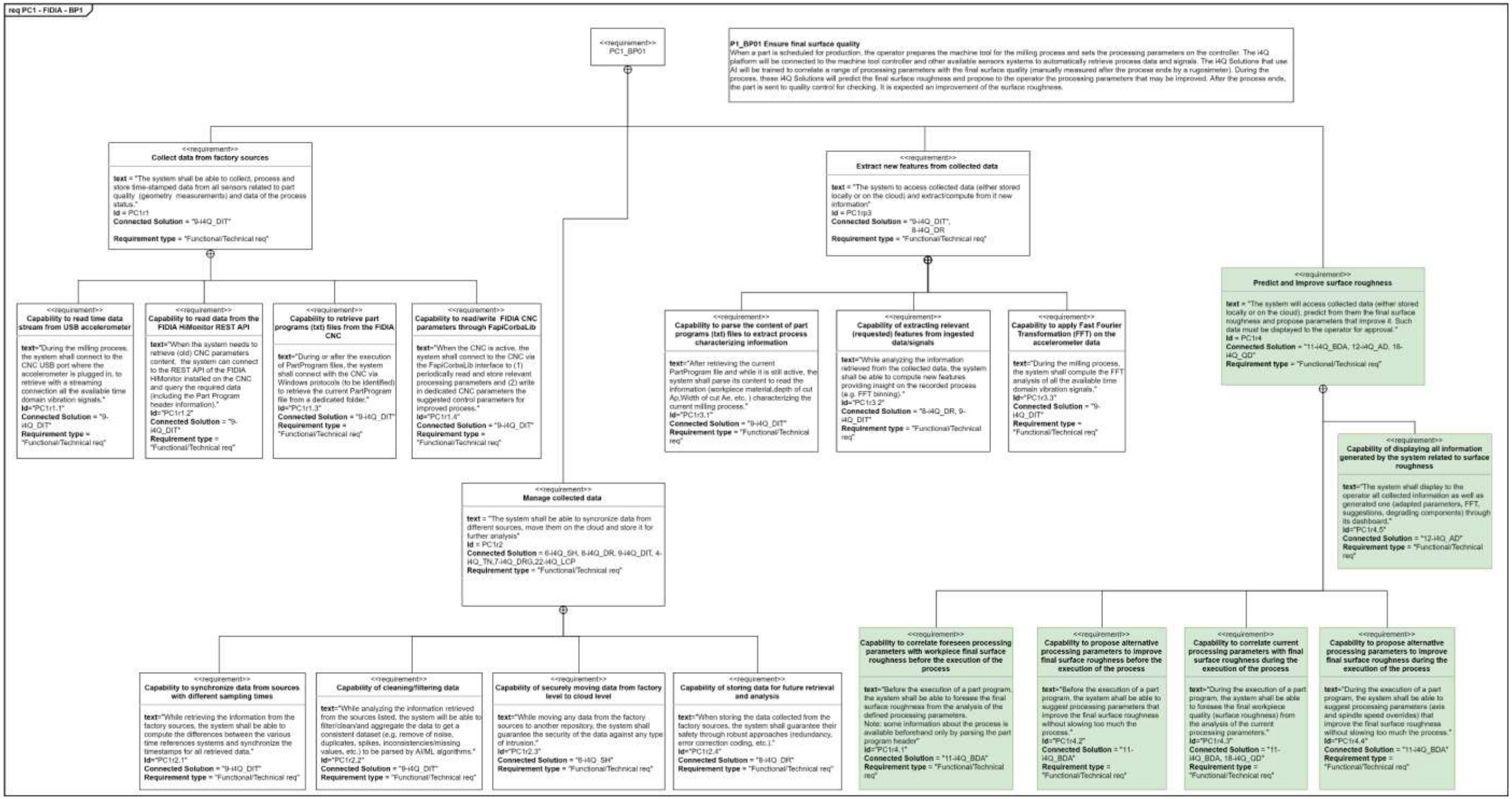


Figure 14. Pilot 1 BP1: Requirements Diagram

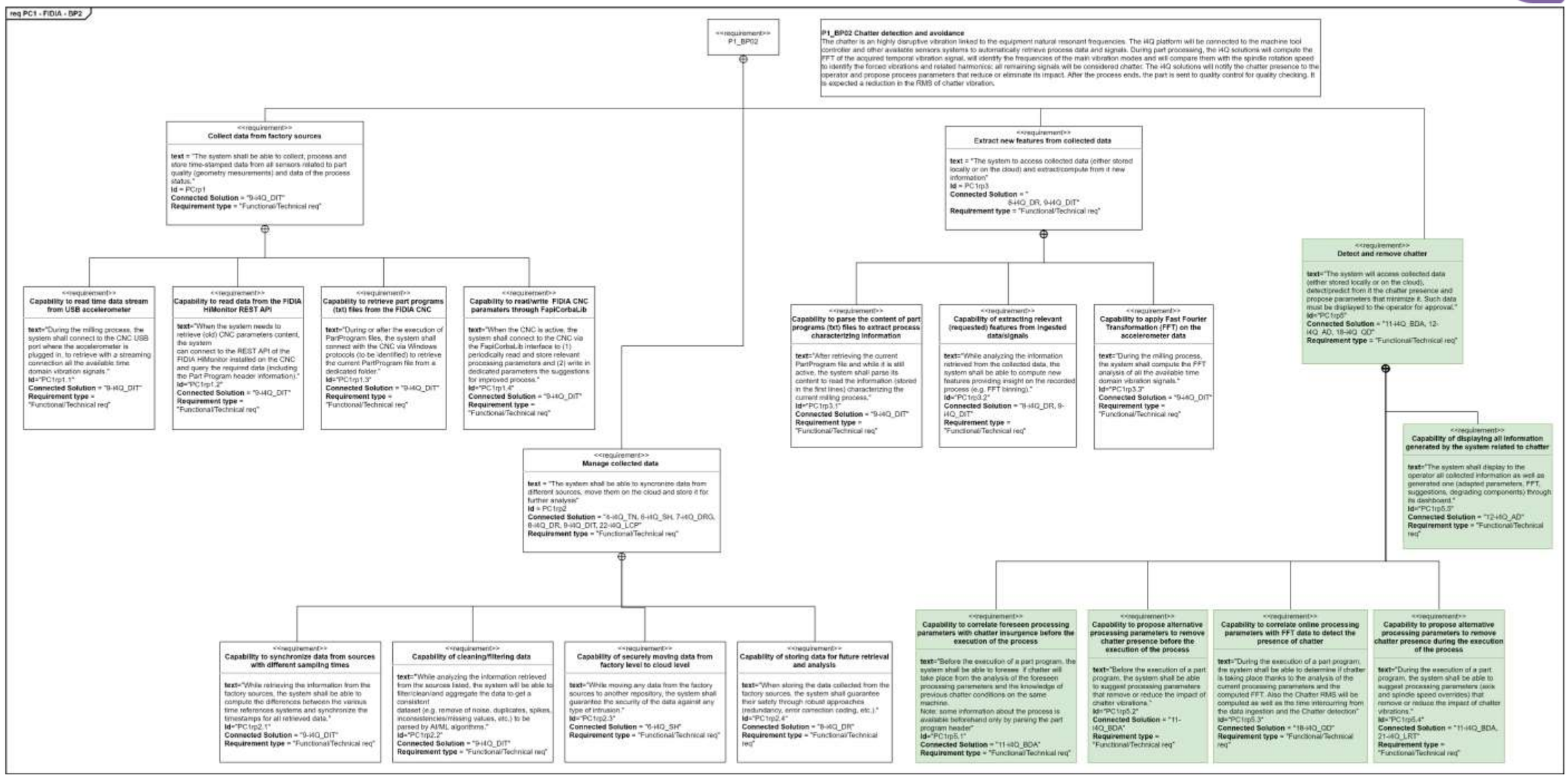


Figure 15. Pilot 1 BP2: Requirements Diagram

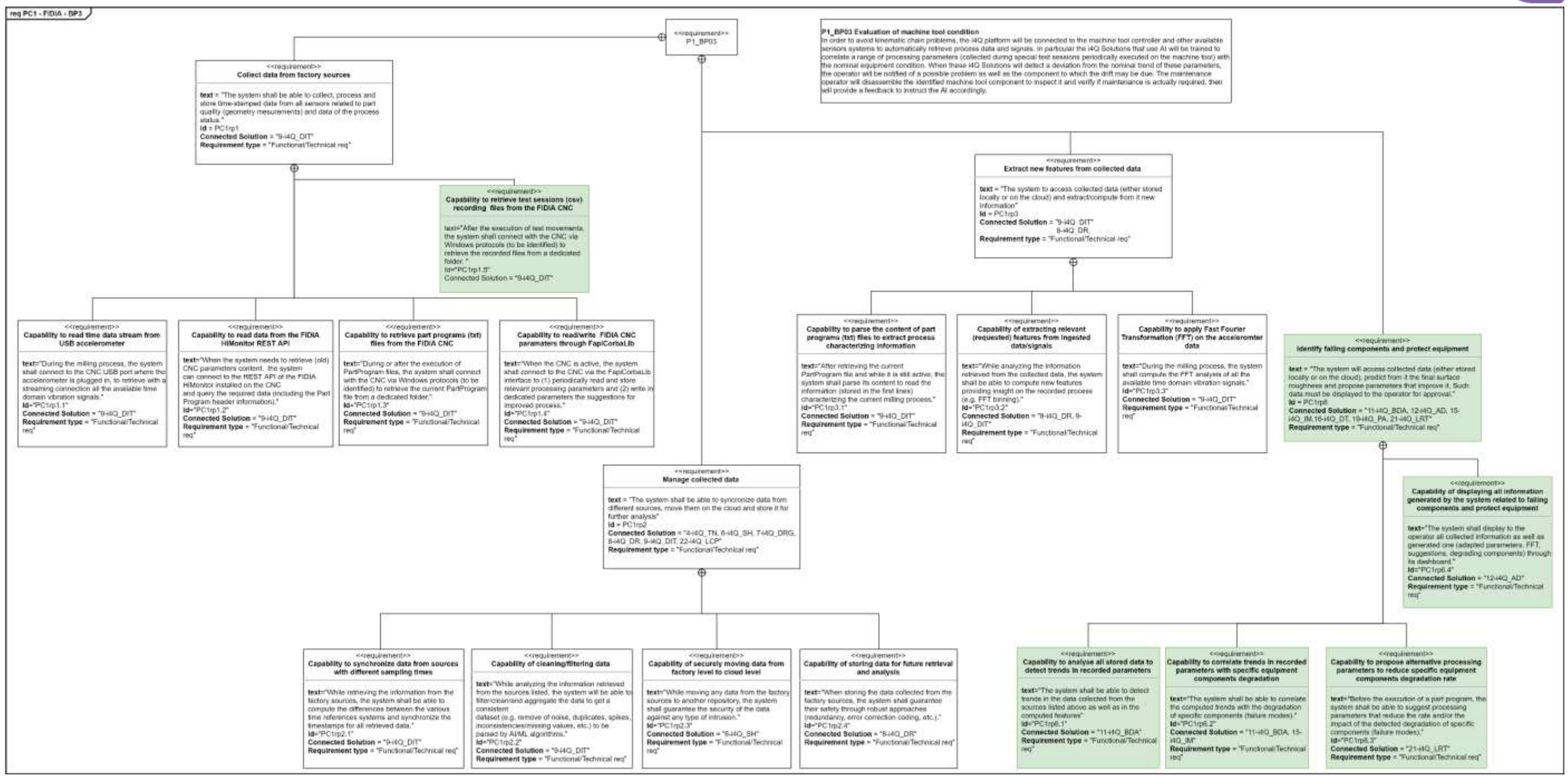


Figure 16. Pilot 1 BP3: Requirements Diagram

Figure 17. Pilot 1: Mapping of (Sub-)Functions to Atomic Requirements

3.1.2 Pilot 2: Diagnostics and IoT Services

Biesse is focusing on including predictive elements which describe the degradation of single machine components or the identification of components which are likelier to break. Therefore, the main challenges that need to be solved are widely spread across all i4Q Solutions. Beginning with feature selection and engineering to several analysis like condition monitoring and ending with the simple visualization of health indicators of the machine and the connection to local systems.

Requirements Diagrams and Mapping of (Sub-)functions to Atomic Requirements

The Pilot Requirement Diagram of the two business processes of Pilot 2 is shown in Figure 18. The requirements are clustered into different main groups. In each requirement box the mapped i4Q Solutions are named.

Figure 19 shows the requirements structure of the business processes of Pilot 2 with the assigned (sub-)functions of the mapped i4Q Solutions. Only the atomic requirements are mapped, since they are the most precise requirements on which the high-level requirements depend. For each defined algorithm it is needed to define machine constraints conditions during the data recording (test cycle), specify the data which should be recorded, the way they are recorded (for example sampling frequency, resolution, etc….) and lastly to define the machine set up to perform test cycle. i4Q Solutions will provide the ability to store and also check this data once the test cycle is executed (PC2r8, PC2r9, PC2r10, PC2r11).

No i4Q solution could be assigned to PC2r15 and PC2r20 (highlighted in yellow), which represents a conflict. Since at this stage, it is not foreseen by i4Q Solutions to provide any user interface to define the test cycles described above, the requirement PC2r20 will be evaluated at a later stage (see Section 5.3).

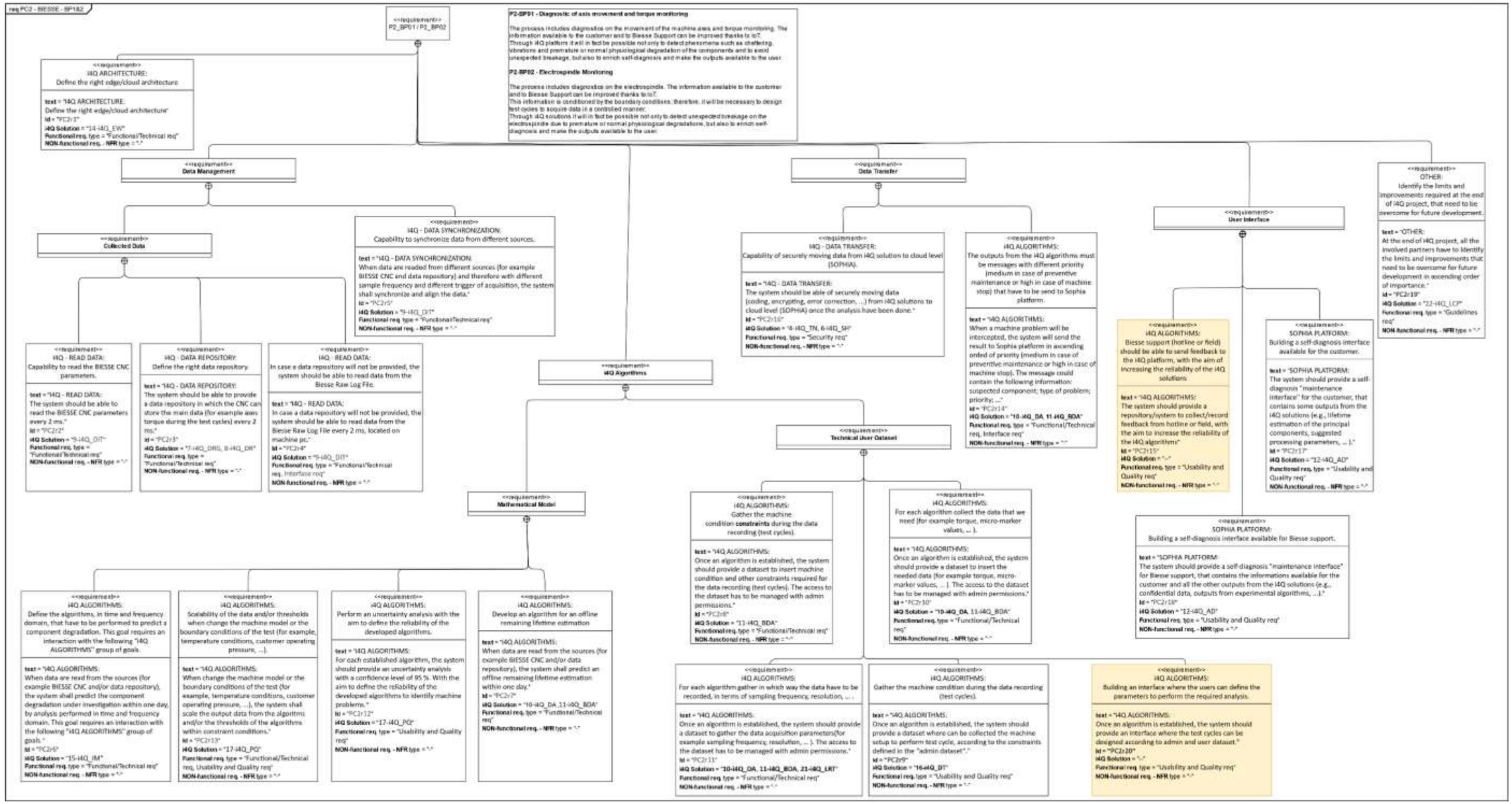


Figure 18. Pilot 2 BP1 and BP2: Requirements Diagram

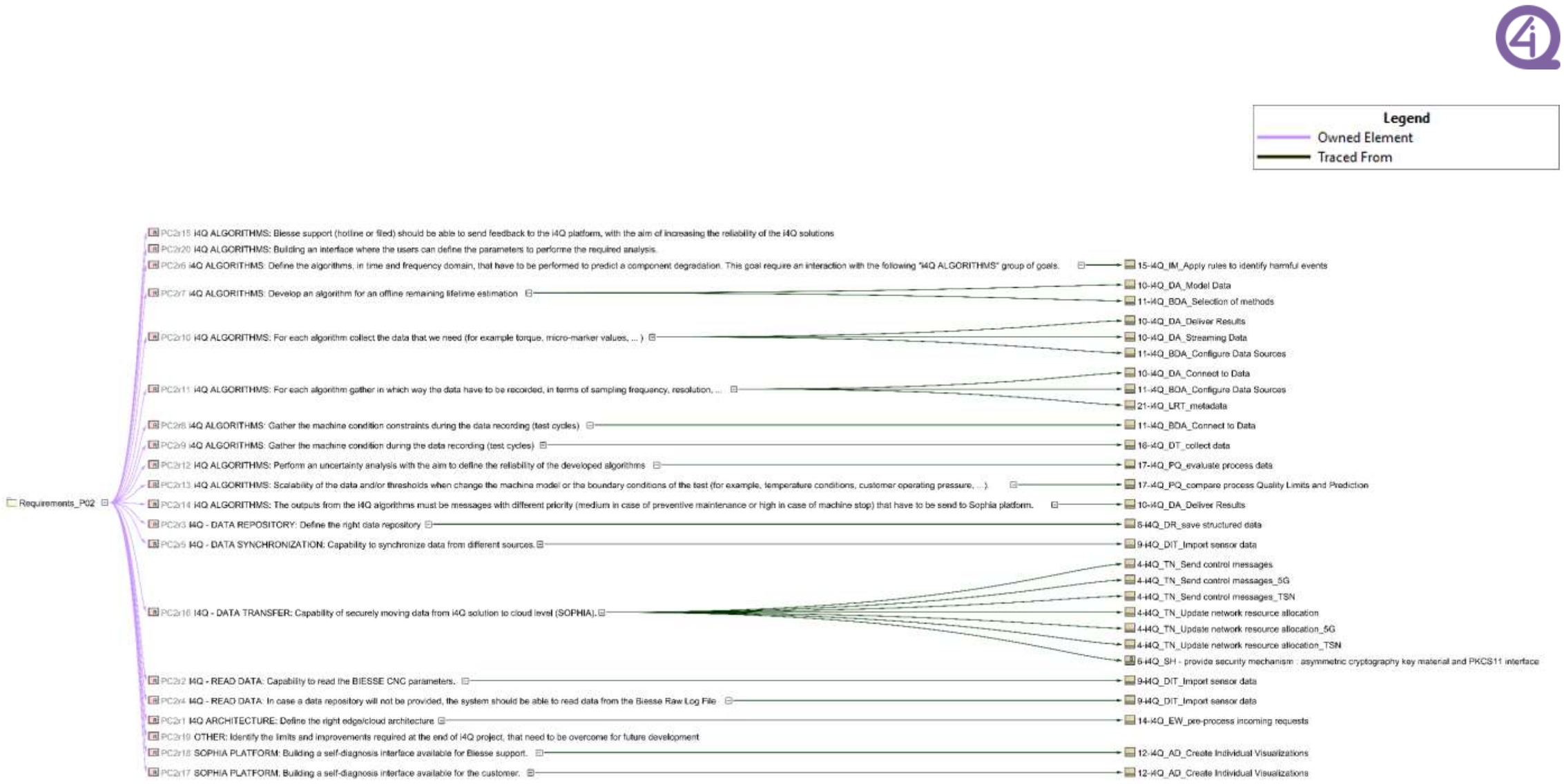


Figure 19. Pilot 2: Mapping of (Sub-)Functions to Atomic Requirements

3.1.3 Pilot 3: White Goods Product Quality

Whirlpool as Pilot 3 is concentrating on the virtualization of their quality tests along the production line. This addresses all solutions which deal with condition monitoring and its evaluation. The final product required is a software package with embedded algorithms in local systems which deals with predictive elements, customized evaluation rules and convenient data visualization.

Requirements Diagrams and Mapping of (Sub-)functions to Atomic Requirements

The Pilot Requirement Diagram of the business process of Pilot 3 is shown in Figure 20. There are four main parent requirements based on the macro-functions i4Q is expected to deliver: A *Predictor*, able to provide a forecast on the product's potential non-conformity, a *Threshold and Importance Analyzer*, to rank the potential issue found and trigger an escalation process; a *Data storage* and a *Data Visualization*. Therein, the mapped i4Q Solutions are named.

Figure 21 shows the requirements structure of the business process of Pilot 3 with the assigned (sub-)functions of the mapped i4Q Solutions. Only the atomic requirements are mapped, since they are the most precise requirements on which the high-level requirements depend. The fulfilment of the requirements by the mapped solution functions can be measured in further steps (see Section 5.3).

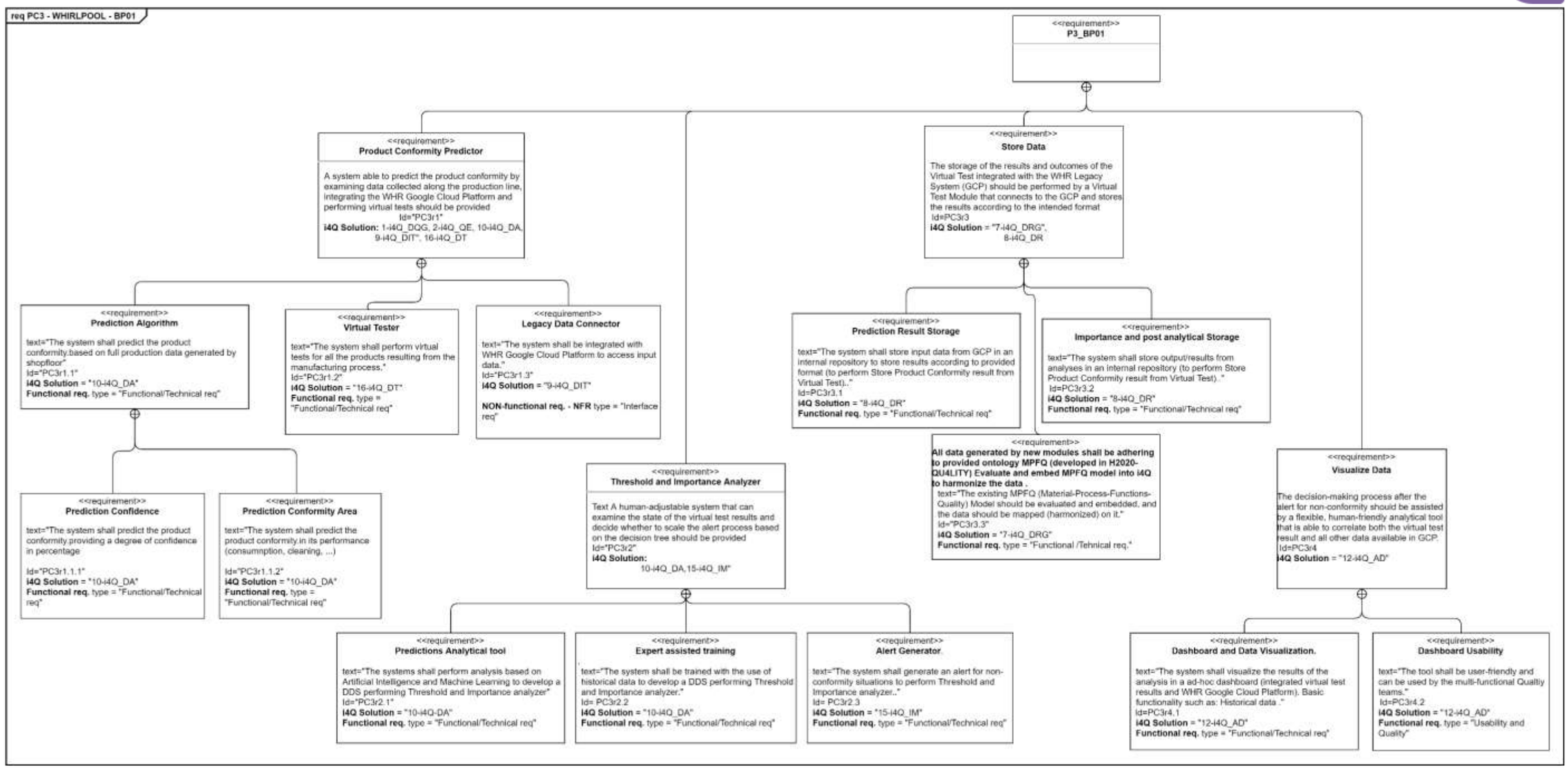


Figure 20. Pilot 3 BP1: Requirements Diagram

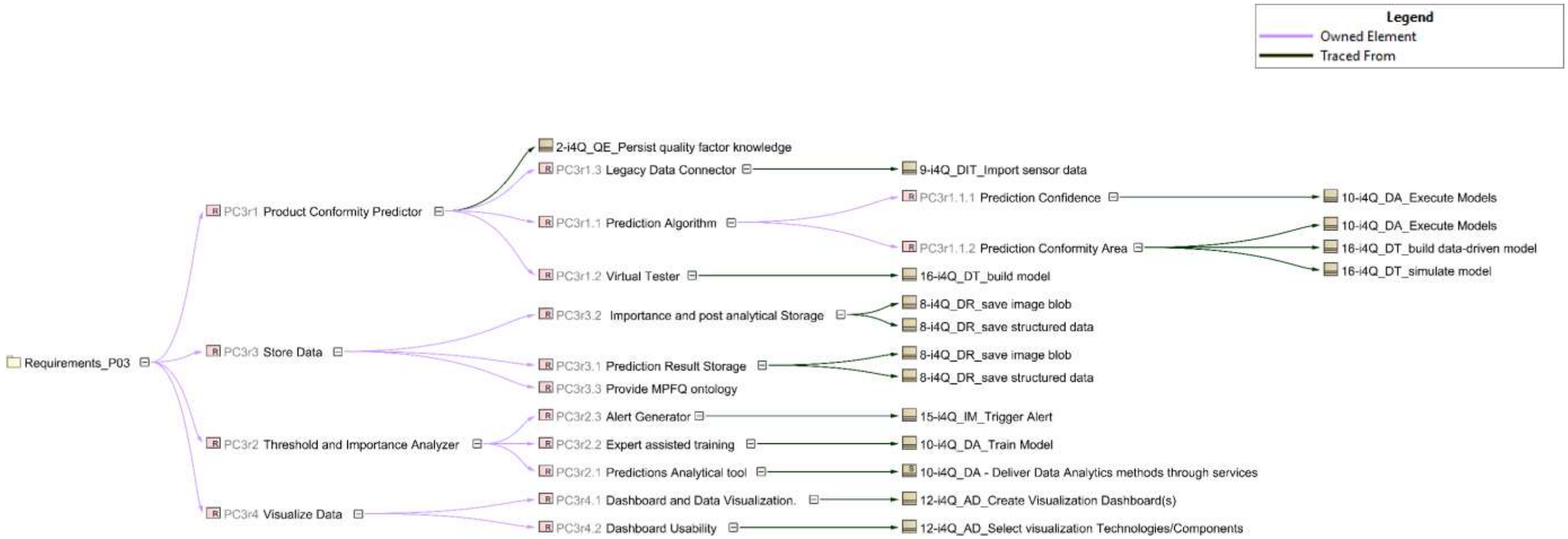


Figure 21. Pilot 3: Mapping of (Sub-)Functions to Atomic Requirements

3.1.4 Pilot 4: Aeronautics and Aerospace Metal Parts Quality

As main focus, FACTOR is dealing with quality management implementations according to Standard AS9100 - Quality Systems - Aerospace - Model for Quality Assurance in Design, Development, Production, Installation and Servicing (AS9100). This covers all the solutions which deal with factory digitization in order to read and collect real-time data, analysis of collected data for product and process deviations, parameter configuration with AI algorithms, and data integration and transformation for manufacturing quality assurance.

Requirements Diagrams and Mapping of (Sub-)functions to Atomic Requirements

The Pilot Requirement Diagrams of the two business processes of Pilot 4 are shown in Figure 22 and Figure 23. Therein, the mapped i4Q Solutions are named. The orange boxes contain requirements that are included in both business processes.

Figure 24 and Figure 25 show the requirements structure of the business processes of Pilot 4 with the assigned (sub-)functions of the mapped i4Q Solutions. Only the atomic requirements are mapped, since they are the most precise requirements on which the high-level requirements depend. The fulfilment of the requirements by the mapped solution functions can be measured in further steps (see Section 5.3).

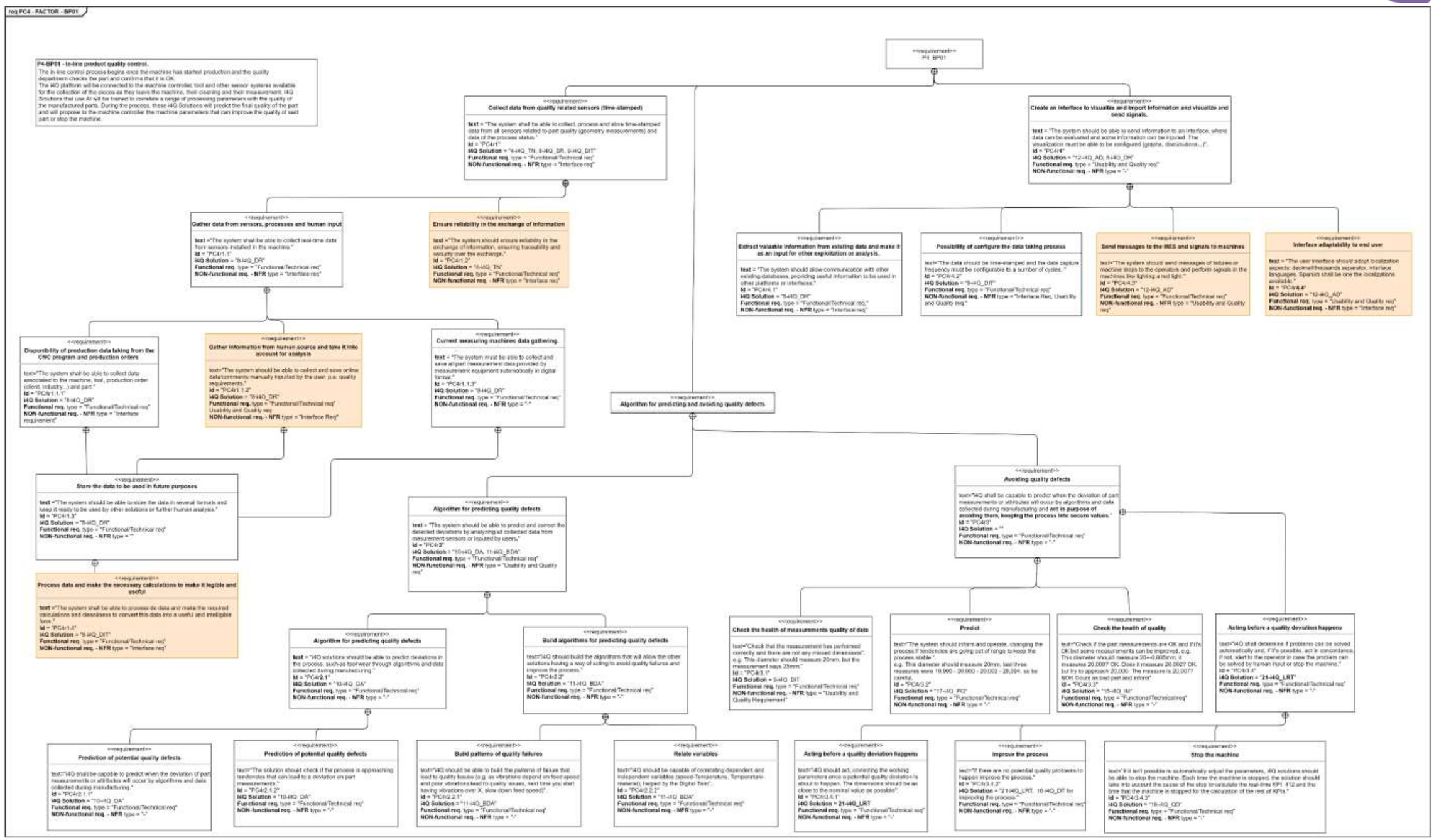


Figure 22. Pilot 4 BP1: Requirements Diagram

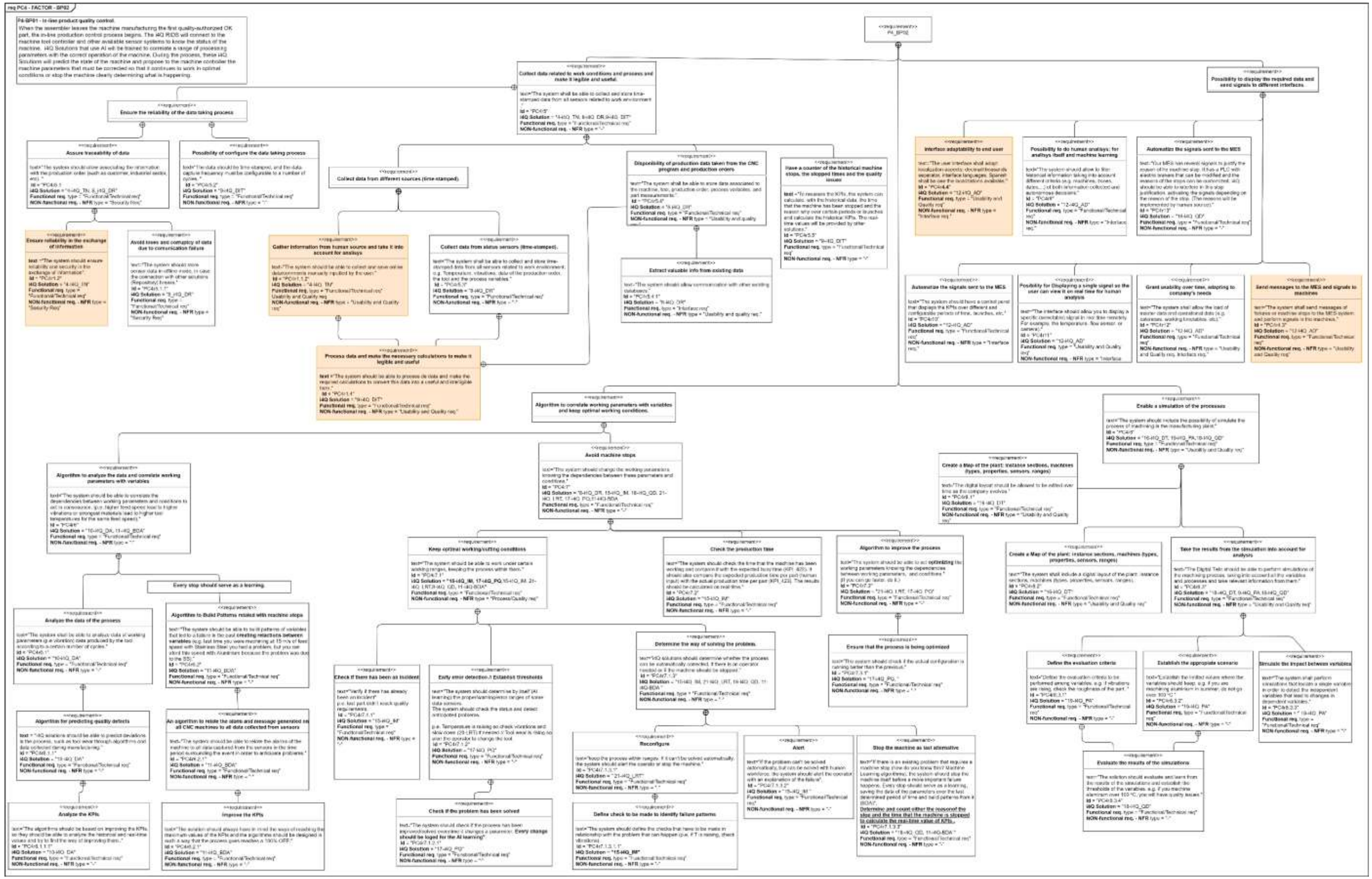


Figure 23. Pilot 4 BP2: Requirements Diagram

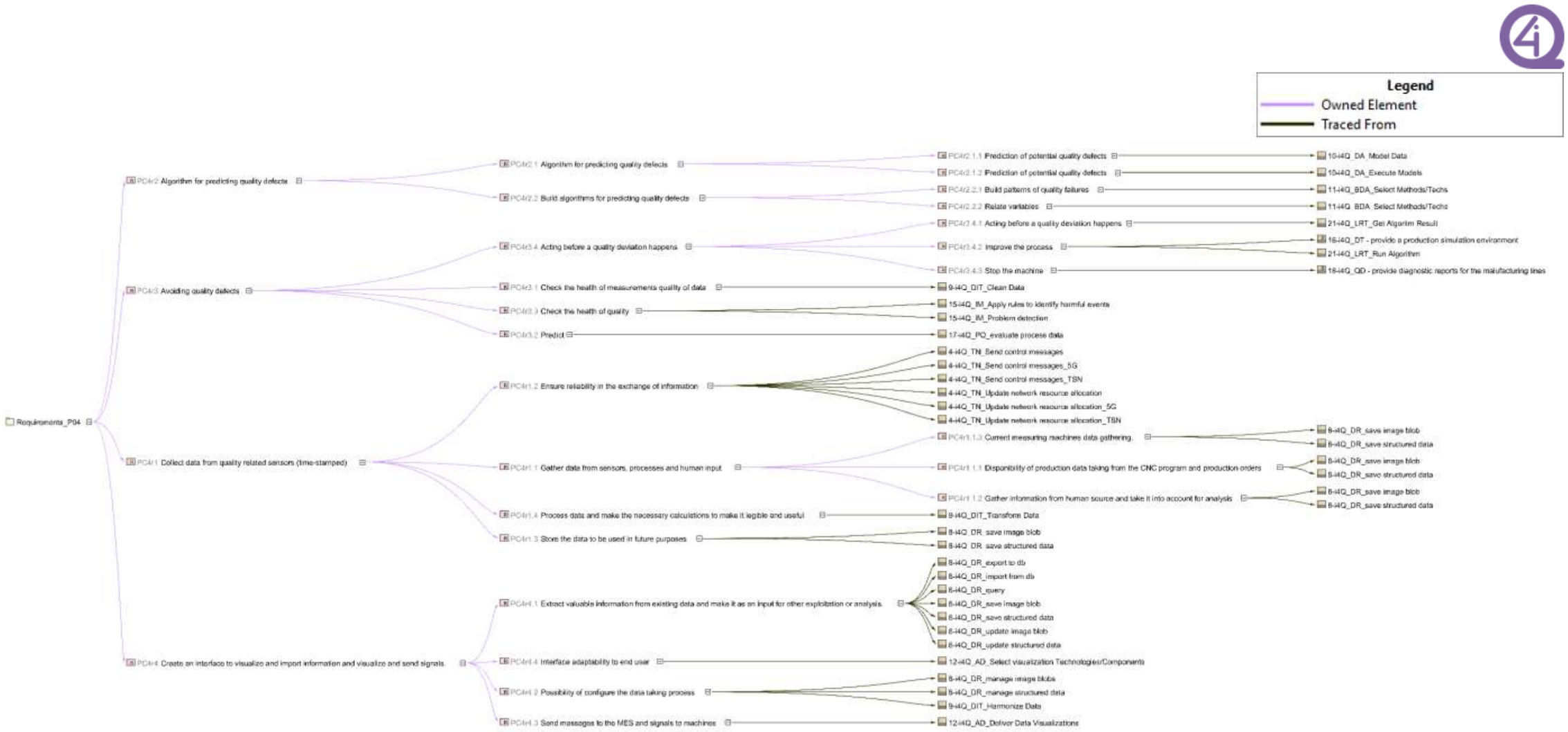


Figure 24. Pilot 4: Mapping of (Sub-)Functions to Atomic Requirements of BP1

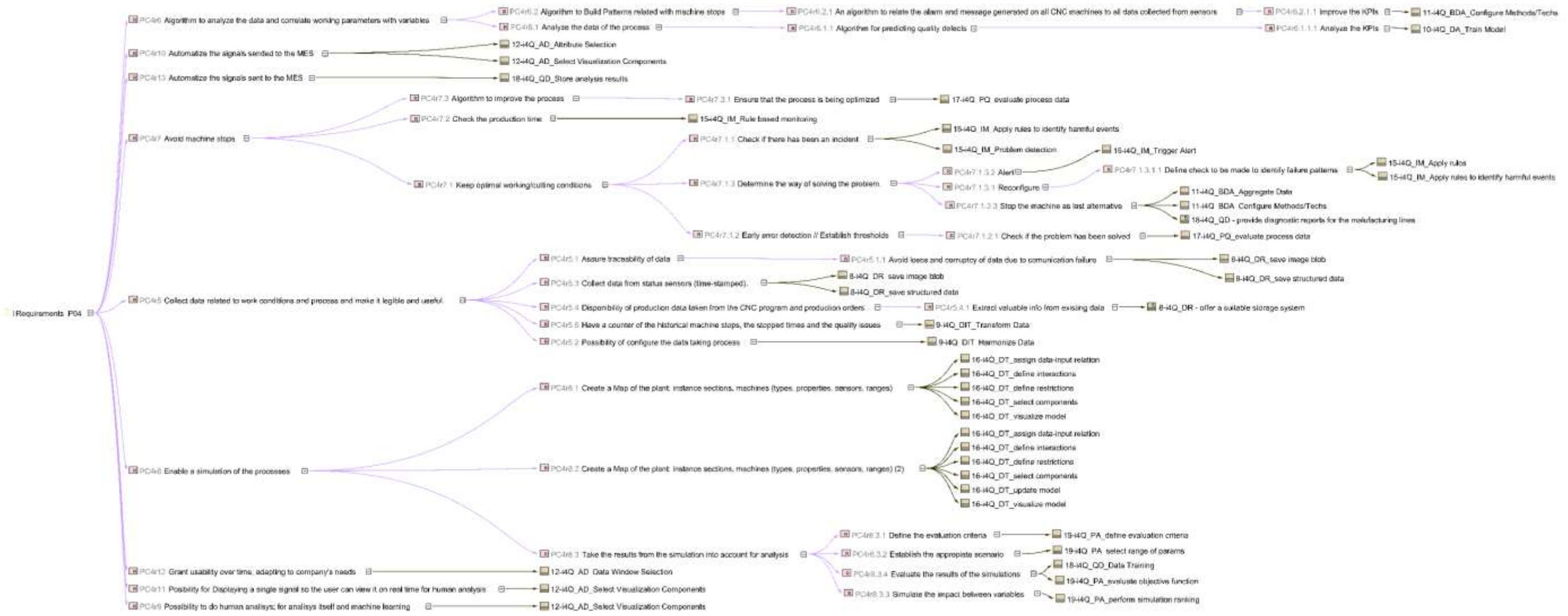


Figure 25.Pilot 4 : Mapping of (Sub-)Functions to Atomic Requirements of BP2

3.1.5 Pilot 5: Advanced In-line Inspection for incoming Prime Matter Quality Control

RiaStone is mainly focusing on in-line, real time inspection and continuous monitoring of the quality conformity of incoming raw matters. Also, it concentrates on the implementation of a single integrated central data analytics system which integrates i4Q data, data originated from installed zero-defect manufacturing system and data from other factory sources. As main challenges, this pilot describes the necessity of implementation of accurate and efficient measuring system and data management tool to integrate and extract business values from collected data.

Requirements Diagrams and Mapping of (Sub-)functions to Atomic Requirements

The Pilot Requirement Diagrams of the two business processes of Pilot 5 are shown in Figure 26 and Figure 27. Therein, the mapped i4Q Solutions are named.

Figure 28 shows the requirements structure of the business processes of Pilot 5 with the assigned (sub-)functions of the mapped i4Q Solutions. Only the atomic requirements are mapped, since they are the most precise requirements on which the high-level requirements depend. The fulfilment of the requirements by the mapped solution functions can be measured in further steps (see Section 5.3).

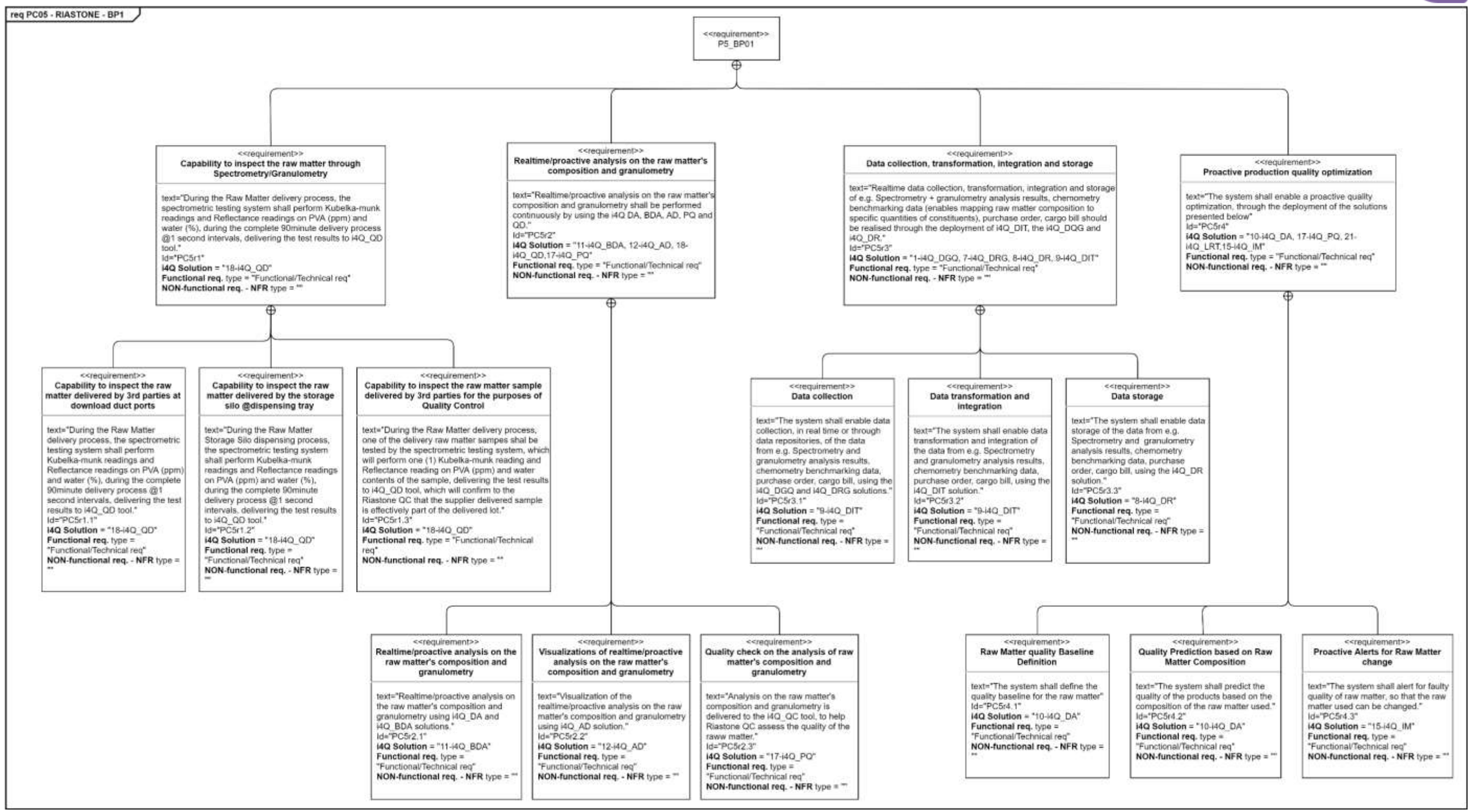


Figure 26. Pilot 5 BP1: Requirements Diagram

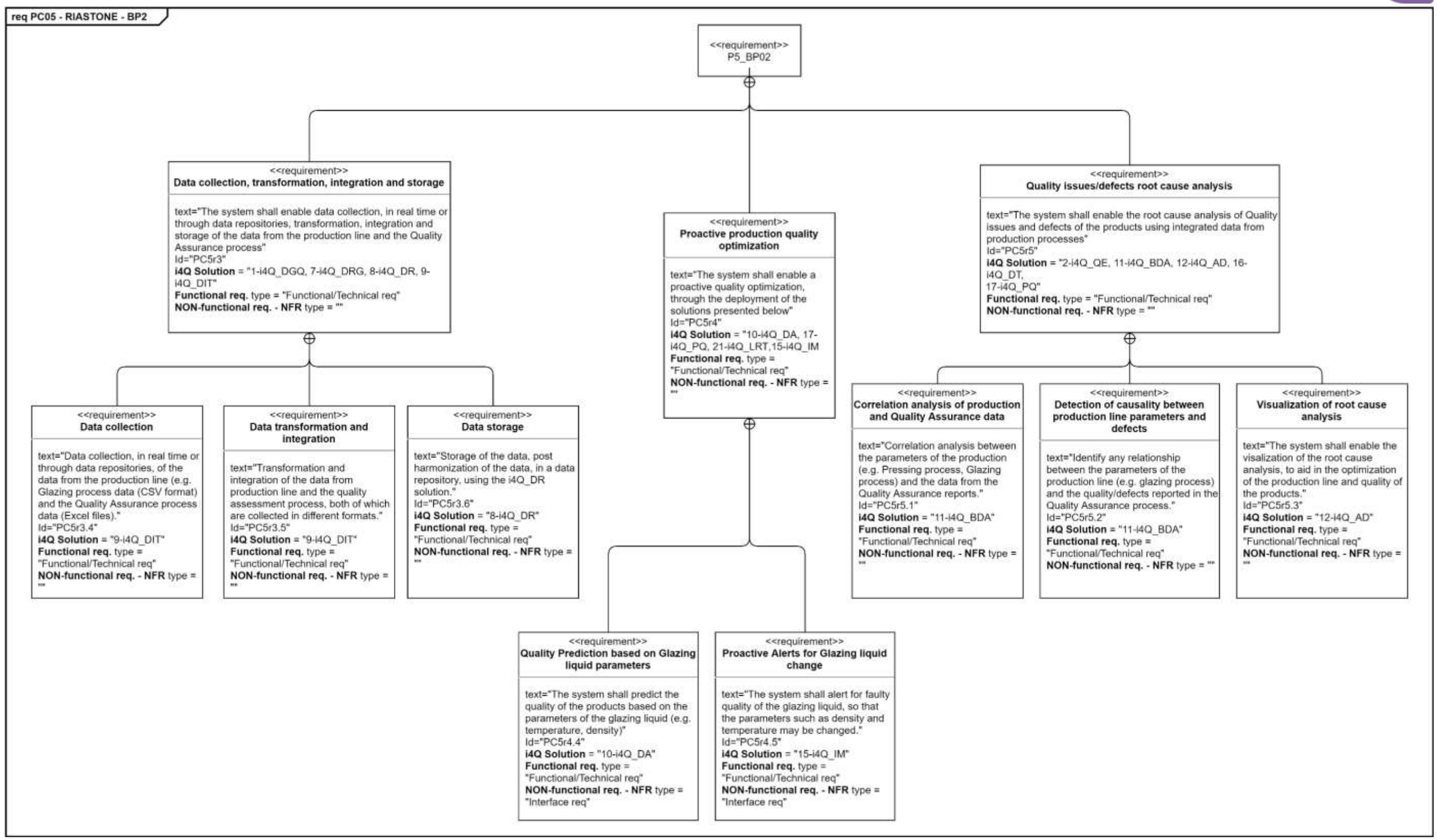


Figure 27. Pilot 5 BP2: Requirements Diagram

Figure 28. Pilot 5: Mapping of (Sub-)Functions to Atomic Requirements

3.1.6 Pilot 6: Automatic Advanced Inspection of Automotive Plastic Parts

Farplas' main objectives are achieving zero-defect in injection moulding production and increase productivity by rapid error identification. The pilot describes the need to learn the process and automatically set the optimized plastic injection moulding machine parameters for each mould/part design and to implement automated quality inspection process and procedure. The main challenges which need to be addressed by i4Q Solutions are standardization of use irrespective of make of moulding machine, sensor equipment, secure data collection, and configuration of parameters to establish inspection condition. The challenges are spread across all i4Q Solutions.

Requirements Diagrams and Mapping of (Sub-)functions to Atomic Requirements

The Pilot Requirement Diagrams of the two business processes of Pilot 6 are shown in Figure 29 and Figure 30. Therein, the mapped i4Q Solutions are named. The diagrams show in white boxes all requirements that apply to all three business processes. The green boxes contain requirements for the specific business process.

Figure 31 shows the requirements structure of the business processes of Pilot 6 with the assigned (sub-)functions of the mapped i4Q Solutions. Only the atomic requirements are mapped, since they are the most precise requirements on which the high-level requirements depend. The fulfilment of the requirements by the mapped solution functions can be measured in further steps (see Section 5.3).

The requirements PC6r6.1, PC6r6.2 and PC6r6.3 conflict with the software solutions, as they are primarily related to hardware functions. Therefore, these requirements are assigned to the quality control hardware *zero gravity 3d*.

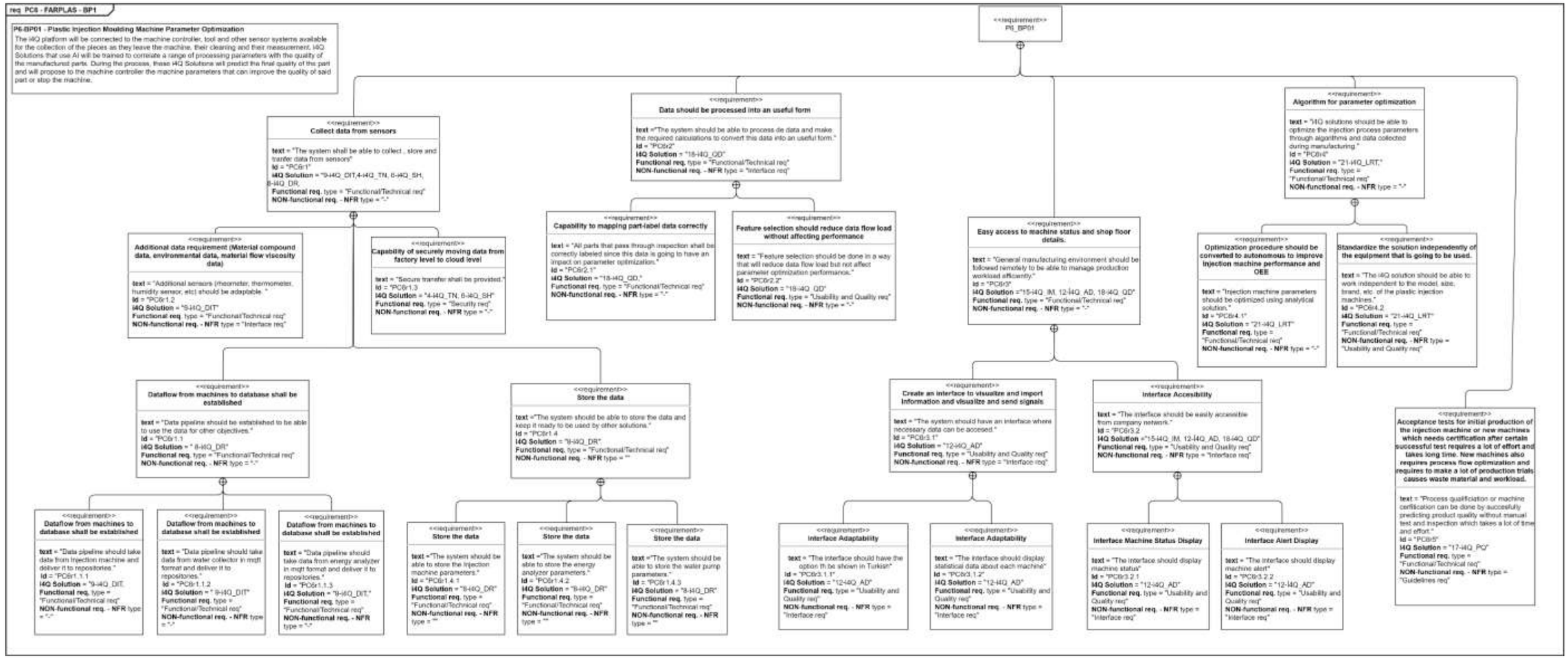


Figure 29. Pilot 6 BP1: Requirements Diagram

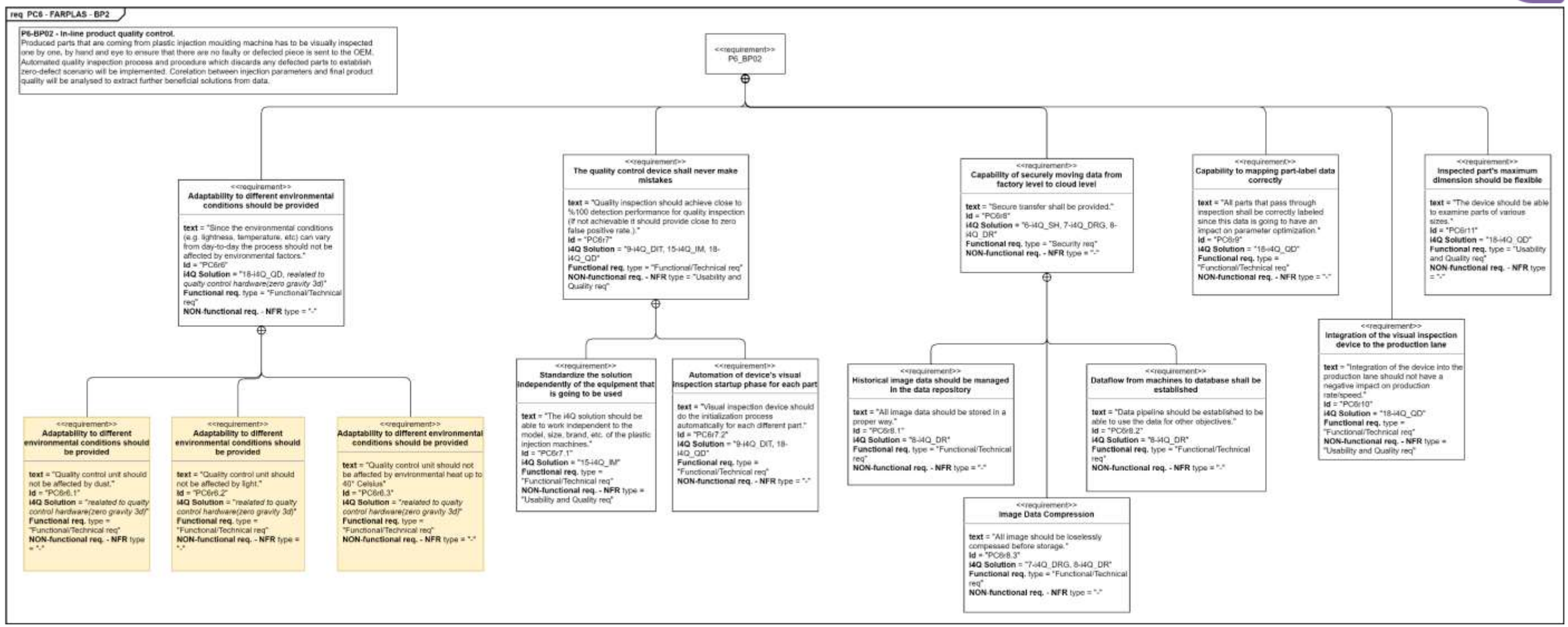


Figure 30. Pilot 6 BP2: Requirements Diagram

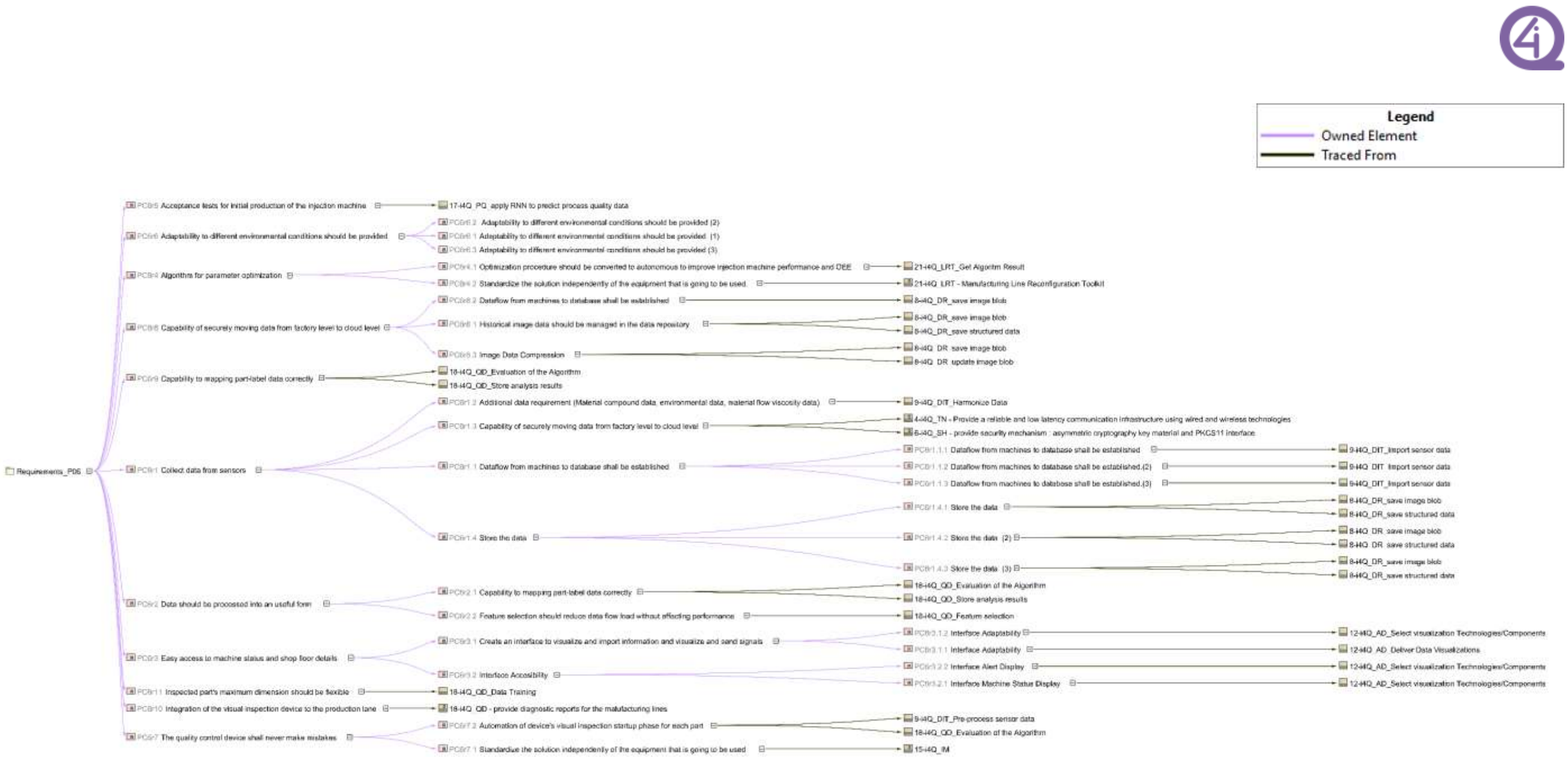


Figure 31. Pilot 6: Mapping of (Sub-)Functions to Atomic Requirements

3.1.7 Pilot 7: Generic Pilot

In work package 6, an additional generic pilot (Pilot 7) will be defined. This pilot will be tested and validated in an experimental facility for i4Q Solutions at the Universitat Politecnica de Valencia (UPV). Additional requirements needed for the development of the i4Q Solutions that have not yet appeared in Pilots 1 to 6 will be identified and described for this Pilot 7. The result will later be included in Deliverable 6.7. In the present Deliverable D1.9, there are no requirements defined for the generic pilot.

3.2 Solution Requirements

In addition to the pilot-specific requirements, the solutions themselves also have requirements, e.g., for the interfaces between the i4Q Solutions. The *Solution Requirement Template* of Deliverable 1.4 with all solution-specific requirements has been updated, revised and extended. The results can be found in the **Appendix I**. The detailed mapping of these requirements to the solution functions is presented in Section 4.

The literature requirements captured in Deliverable 1.4 are on a very generic level, so they are not considered in the detailed mapping of the solution functions in Section 4.

4. Functional Specification

The functions of the i4Q Solutions have been roughly described in D1.4. In this section, a deeper description and functional specification of the i4Q Solutions is now given through the Function Structure Diagrams (FSD) and mapped requirements for each solution. Each of the subsections provides information on the following three parts.

(1) First, the Functional Structure Diagram (FSD) of each i4Q Solution is presented. These FSD have been modelled with the open-source software tool diagrams.net[8] (former draw.io) according to the method described in Section 2.3.3. The detailed functional specification with description of the input, output and functional elements of the FSD can be found in *D2.5 Functional Specification*.

(2) In the second part, the mapping of all gathered requirements to the solution and its (sub-)functions is described by diagrams according to the method described in Section 2.3.5. These diagrams have been created using the Cameo System Modeler software tool[9]. Included are the complete sets of requirements collected from the pilots and the solution providers. This considers the knowledge and expertise of the partners concerning all pilots and solution interfaces.

    The requirements can be all defined as *functional requirements,* so the types of requirements are clustered in

    - *Requirements* → Application requirements that are defined by the pilot cases.
    - *Extended Requirements* → Interface requirements that are defined by the solution interfaces.

(3) In the third part, the results are analysed and discussed according to the evaluation criteria *Completeness*, *Precision*, *Interface specification,* and *Req Origin* that are described in Section 2.3.7. The most important aspects and critical points are outlined as input for the following work packages. The complete overview of the *Functional Specification Analysis Results* can be found in **Appendix III**.

The guideline solutions are briefly described in the corresponding requirement tables. For the Manufacturing Line Data Certification Procedure (22-i4Q_LCP) a FSD is created and the requirements are mapped in the diagram.

[8] [8] https://app.diagrams.net/

[9] [9] https://www.3ds.com/products-services/catia/products/no-magic/cameo-systems-modeler/

4.1 i4Q^{DQG} Data Quality Guidelines

i4Q^{DQG} is defined in Deliverable 1.4 as a document that contains guidelines about data quality management in production, including a "terminology and definitions" section to clarify key terms. The Data Quality Guideline is based on contents identified within the context of the pilot use cases, and the i4Q Solutions. Relevant aspects to consider are data and information flows, relevant quality characteristics identified by partners, relevant factors that influence data and information quality, existing measures to manage data quality-related factors, and existing strategies to apply measures. These inputs will be complemented with related contents in literature.[10]

After a revision of all requirements in this second deliverable of the requirements analysis and functional specification task, the following requirements are set up for the i4Q^{DQG} (Table 2).

ID	Type of Requirement	Requirement Title	Requirement definition (related to the i4Q Solution)	Requirement coming from connected Solution
01BIBAr1	Usability and Quality req	Easy to operationalize	The guideline should be easy to operationalize.	1-i4Q_DQG
01BIBAr2	Guidelines req	Cover long-term and short-term measures	The guideline should cover long-term and short-term measures to improve data quality.	1-i4Q_DQG
01BIBAr3	Guidelines req	Focus on information	The guideline should focus on information, not on database quality improvements.	1-i4Q_DQG
01BIBAr4	Guidelines req	Use data life cycle model	The guideline should use a data life cycle model to define its scope.	1-i4Q_DQG

Table 2. Requirements mapped to i4Q^{DQG}

4.2 i4Q^{QE} QualiExplore for Data Quality Factor Knowledge

i4Q^{QE} is defined in Deliverable 1.4 as a small software tool to organize and visualize factors that influence data quality. It grounds on the *Evolutional Data Quality* approach suggested by Liu

[10] This description is mainly taken from i4Q Deliverable 1.4 (Nowak-Meitinger and Jochem, 2021, p. 62f).

and Chi (Liu, L. and Chi, L., 2002): Collection Quality > Organization Quality > Presentation Quality > Application Quality.[11]

The figures below provide the following information:

(1) Figure 32 shows the FSD of i4Q^{QE} with its input and outputs as well as the functions and sub-functions. Interfaces to other i4Q Solutions are also included. This provides a structural overview of the functionalities of the solution.
(2) Figure 33 shows the mapped requirements to the solution functions. All pilot and solution requirements are included. This provides an overview of the current level of completeness of the requirements analysis and functional specification of the solution.
(3) The mapping results are analysed and discussed according to the evaluation criteria described in Section 2.3.7:
    - ***Completeness*** – The mapping diagram shows that 5 out of 12 sub-functions (lowest level) have mapped requirements, which corresponds to a completeness value of 0.42. The "blank spots" where no requirements are mapped in the diagrams could carry the risk of (still) unknown requirements not being fulfilled, while at the same time giving developers room to act because no specific requirements are given in the use case scenarios.
    - ***Precision*** – The mapping diagram shows that 4 out of 9 requirements have been mapped to sub-functions (lowest level), which corresponds to a precision value of 0.44. It should be checked, if further specification by users (pilots) and solution providers is needed to be more precise in the current requirements descriptions and their mapping.
    - ***Interface specification*** – In the FSD zero interfaces are identified. Since 22-i4Q_LCP will interact with 2-i4Q_QE this interface should be added and clarified in the upcoming work packages 3 and 5 to ensure that the solutions can cooperate.
    - ***Req Origin*** – The requirements have been defined by 3 different stakeholders. These should be considered in the BUILD phase as well as the developers' team of 22-i4Q_LCP.

The missing requirements for certain sub-functions exist because these functions are mandatory technical functions resulting from certain features. For example, authenticating and authorizing users typically requires token management, or user interactions require reading and writing in the database. These requirements are not included because the focus is on new aspects introduced in i4Q.

The procedure for creating the diagrams and the purpose is described in Section 2. Further descriptions of the solution can be found in Deliverable 2.7 as well as in Deliverable 1.4. The functions of the solution are described in detail in Deliverable 2.5.

[11] This description is mainly taken from i4Q Deliverable 1.4 (Nowak-Meitinger and Jochem, 2021, p. 65f).

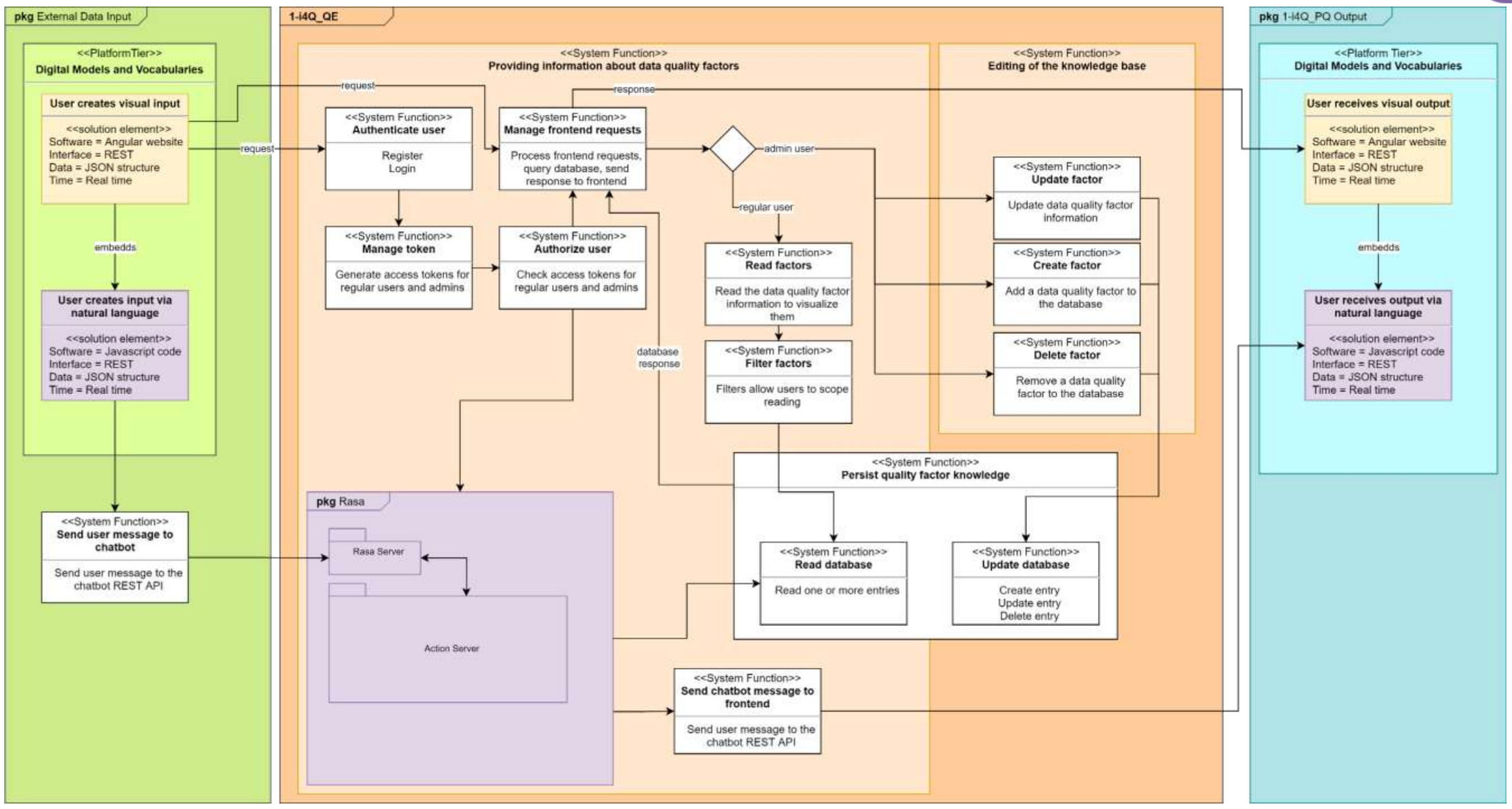


Figure 32. i4Q^{QE} Function Structure Diagram (FSD)

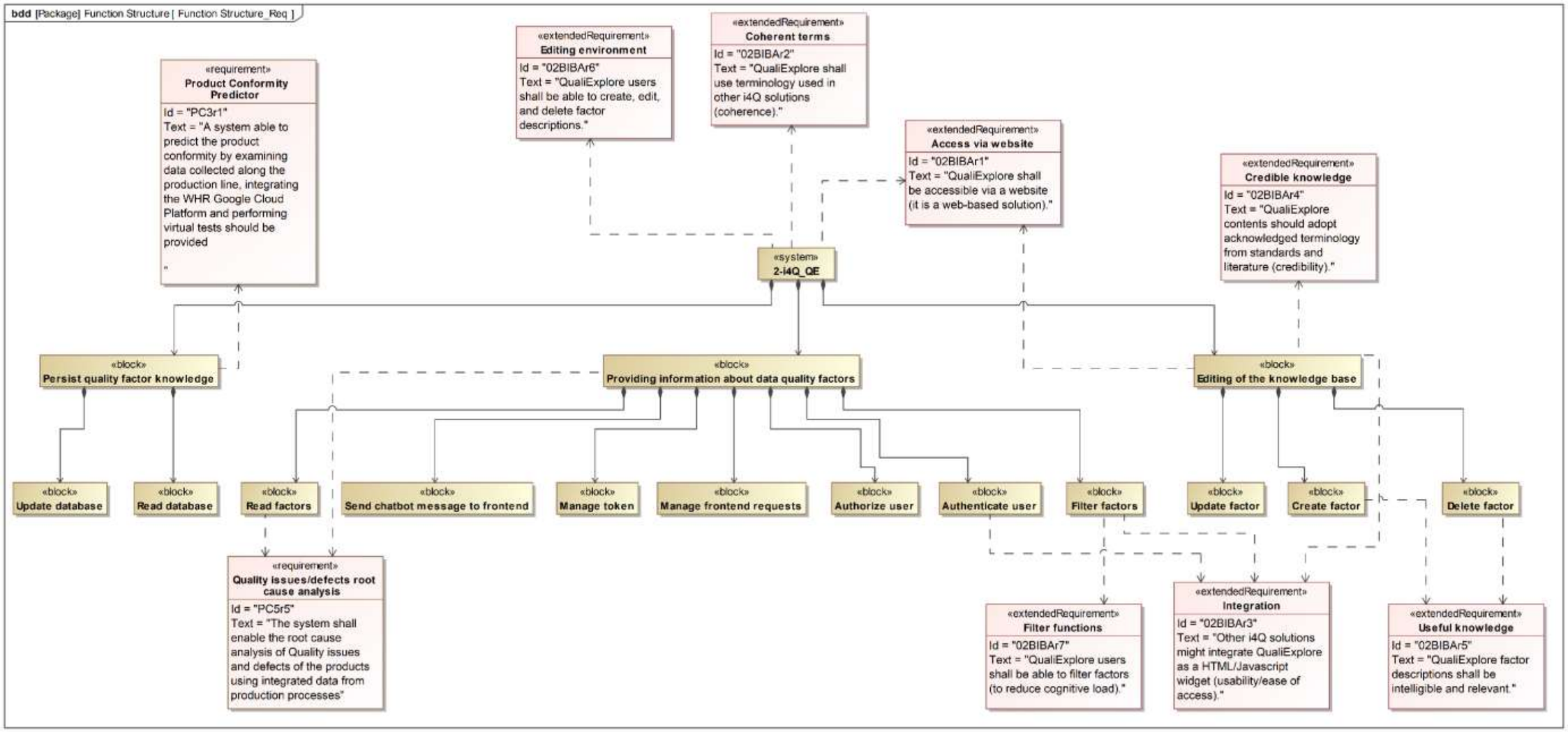


Figure 33. i4Q[QE] Requirements Mapping and Functional Specification

4.3 i4Q^{BC} Blockchain Traceability of Data

i4Q^{BC} is defined in Deliverable 1.4 as a solution that provides easy, trusted and traceable access to data coming from many different sources. It shall enhance the level of trust in the platform by employing a blockchain based data service, to support data traceability in the data that flows directly to the blockchain, thus serving as a single point of truth, preserving provenance and supporting non-repudiation.[12]

The figures below provide the following information:

(1) Figure 34 shows the FSD of i4Q^{BC} with its input and outputs as well as the functions and sub-functions. Interfaces to other i4Q Solutions are also included. This provides a structural overview of the functionalities of the solution.
(2) Figure 35 shows the mapped requirements to the solution functions. All pilot and solution requirements are included. This provides an overview of the current level of completeness of the requirements analysis and functional specification of the solution.
(3) The mapping results are analysed and discussed according to the evaluation criteria described in Section 2.3.7:
    - ***Completeness*** – The mapping diagram shows that 0 out of 5 sub-functions (lowest level) have mapped requirements, which corresponds to a completeness value of 0.0. The "blank spots" where no requirements are mapped in the diagrams could carry the risk of (still) unknown requirements not being fulfilled, while at the same time giving developers room to act because no specific requirements are given in the use case scenarios. The integration and use of the BC solution still seems abstract, which makes it difficult for partners to define requirements. This should be clarified in the BUILD work packages.
    - ***Precision*** – The mapping diagram shows that 0 out of 2 requirements have been mapped to sub-functions (lowest level), which corresponds to a precision value of 0.0. It should be checked, if further specification by users (pilots) and solution providers is possible in order to specify the current requirements descriptions and their mapping more precisely.
    - ***Interface specification*** – In the FSD 2 input interfaces are identified. Since 3-i4Q_BC should be connected to several i4Q Solutions these interfaces should be added and clarified in the upcoming work packages 3 to 5 to ensure the interoperability.
    - ***Req Origin*** – The requirements have been defined by 1 stakeholder. To ensure that this solution is created in such a way that many other i4Q Solutions can use it, collaboration with all affected stakeholders should be considered.

[12] This description is mainly taken from i4Q Deliverable 1.4 (Nowak-Meitinger and Jochem, 2021, p. 68f).

The procedure for creating the diagrams and the purpose is described in Section 2. Further descriptions of the solution can be found in Deliverable 2.7 as well as in Deliverable 1.4. The functions of the solution are described in detail in Deliverable 2.5.

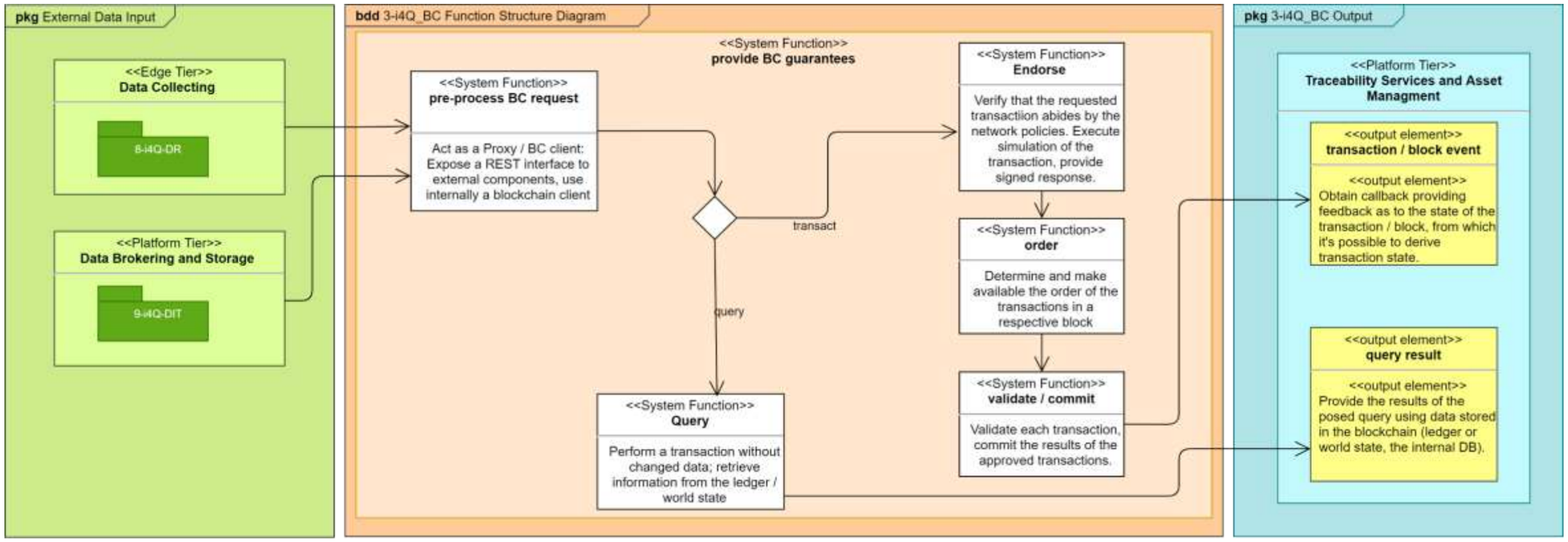


Figure 34. i4Q[BC] Function Structure Diagram (FSD)

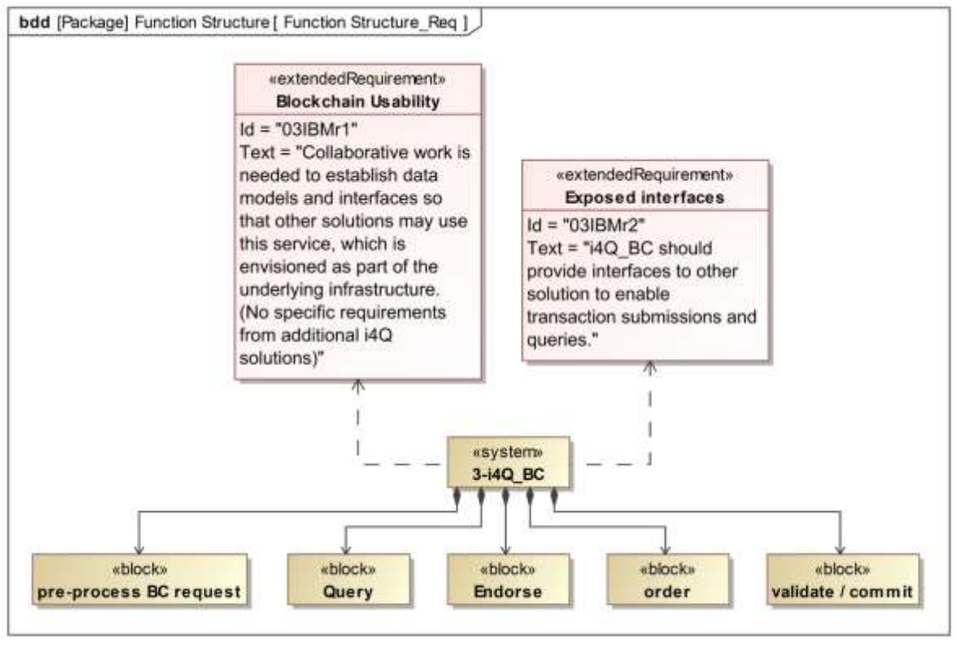


Figure 35. i4Q[BC] Requirements Mapping and Functional Specification

4.4 i4Q^{TN} Trusted Networks with Wireless and Wired Industrial Interfaces

i4Q^{TN} is defined in Deliverable 1.4 as a group of communication technologies such as TSN for wired communications, and wireless access networks (e.g., Wireless Sensors Technologies (WSN), low-power wide-area network (LPWAN), ad-hoc connections) configured and optimized to improve reliability of the communication infrastructure and therefore the integrity and reliability of data collected in the floor plant. Depending on the use case requirements, i4Q^{TN} provides the robust and cost-effective solution to connect devices, sensors, and machines to other management, data analysis, storage, or visualization systems that need this collected data.[13]

The figures below provide the following information:

(1) Figure 36 shows the FSD of i4Q^{TN} with its input and outputs as well as the functions and sub-functions. Interfaces to other i4Q Solutions are also included. This provides a structural overview of the functionalities of the solution.
(2) Figure 37 shows the mapped requirements to the solution functions. All pilot and solution requirements are included. This provides an overview of the current level of completeness of the requirements analysis and functional specification of the solution. The complete requirements mapping to i4Q^{TN} is also included in the matrix in **Appendix II**.
(3) The mapping results are analysed and discussed according to the evaluation criteria described in Section 2.3.7:
    - ***Completeness*** – The mapping diagram shows that 6 out of 20 sub-functions (lowest level) have mapped requirements, which corresponds to a completeness value of 0.3. The "blank spots" where no requirements are mapped in the diagrams could carry the risk of (still) unknown requirements not being fulfilled, while at the same time giving developers room to act because no specific requirements are given in the use case scenarios.
    - ***Precision*** – The mapping diagram shows that 9 out of 12 requirements have been mapped to sub-functions (lowest level), which corresponds to a precision value of 0.75. It should be checked, if further specification by users (pilots) and solution providers is needed in the current requirements descriptions and their mapping.
    - ***Interface specification*** – In the FSD 3 output interfaces are identified and defined. Since 4-i4Q_TN should be connected to several i4Q Solutions these interfaces should be added and clarified in the upcoming work packages 3 to 5 to ensure the interoperability.

[13] This description is mainly taken from i4Q Deliverable 1.4 (Nowak-Meitinger and Jochem, 2021, p. 70).

- *Req Origin* – The requirements have been defined by 6 different stakeholders. To ensure that this solution is created in such a way that many other i4Q Solutions can use it, collaboration with all affected stakeholders should be considered.

The low value of completeness criteria has several reasons. First, sub-functions related to the orchestration of different communication technologies (14 of the 20 sub-functions) have a requirement mapped to a higher-level block function (Full network orchestration), because of the early stage of the solution. Second, some sub-functions (associated with Translate the controller specification in generic network configuration) are included in the FSD in order to provide connectivity with generic networks, such that other partner/solutions are able to interoperate with Trusted Networks solution. At the moment, there is no requirements assigned for these sub-functions. With the development of the solution, these requirements will be more specific and both criteria will be improved.

The procedure for creating the diagrams and the purpose are described in Section 2. Further descriptions of the solution can be found in Deliverable 2.7 as well as in Deliverable 1.4. The functions of the solution are described in detail in Deliverable 2.5.

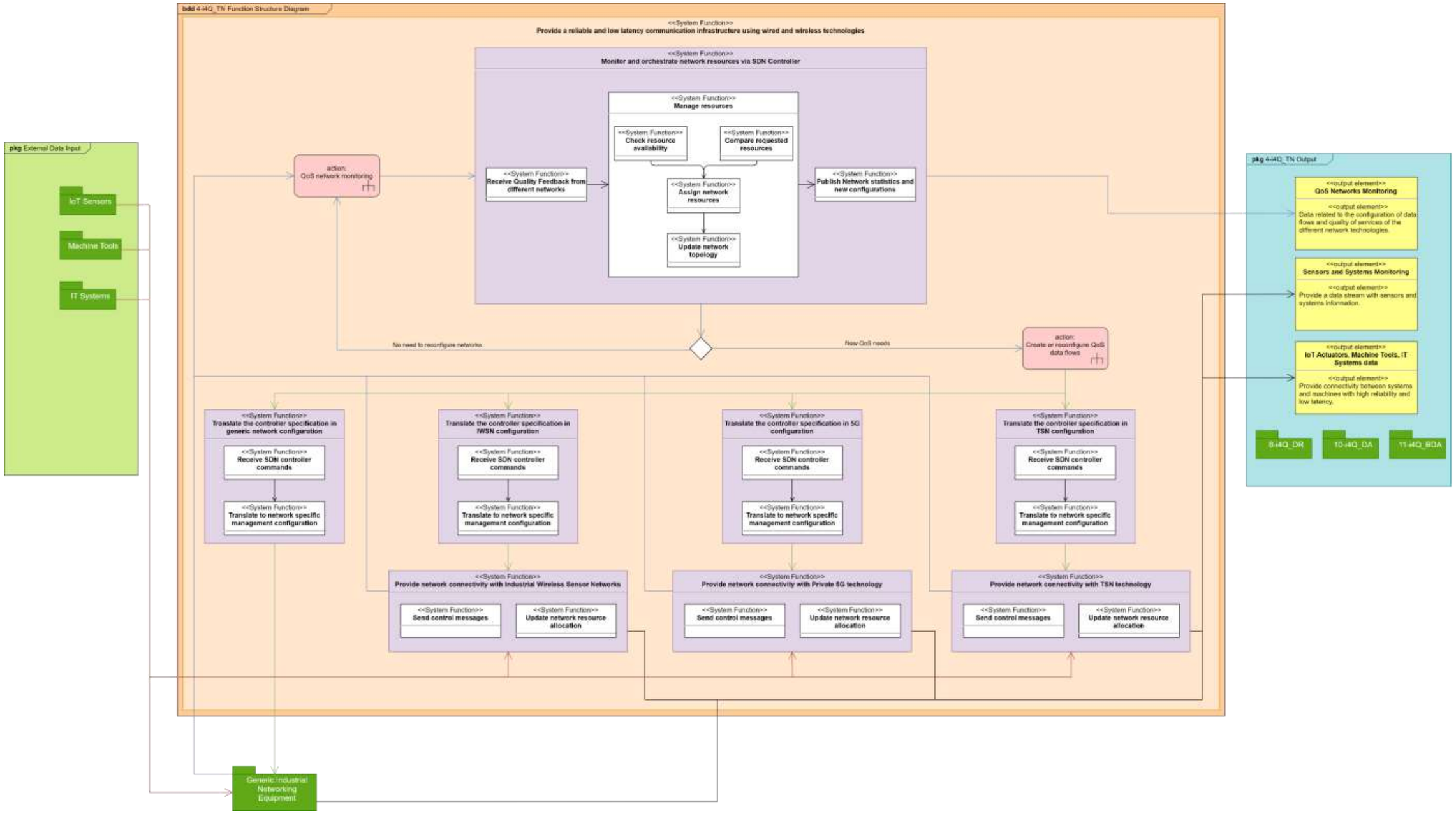


Figure 36. i4Q^{TN} Function Structure Diagram (FSD)

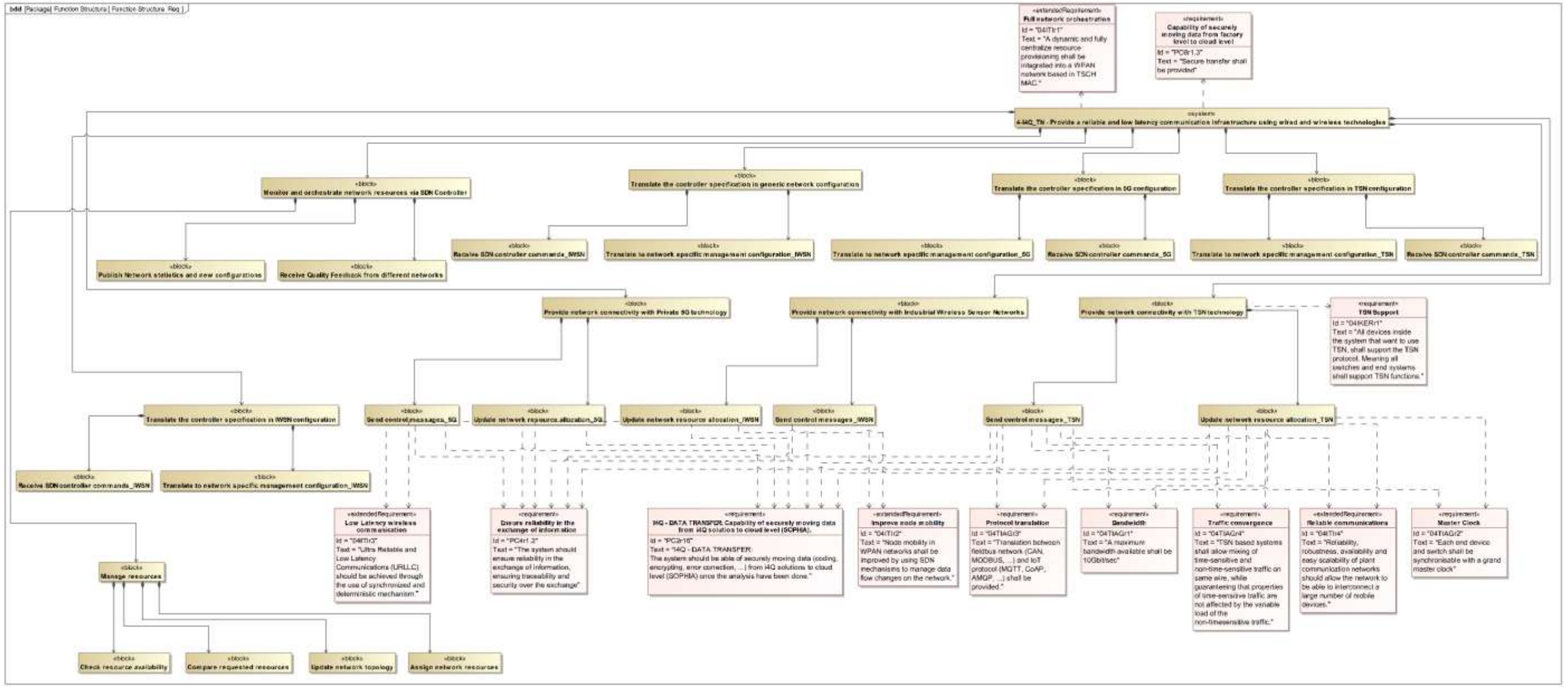


Figure 37. i4Q^{TN} Requirements Mapping and Functional Specification

4.5 i4Q^{CSG} Cybersecurity Guidelines

i4Q^{CSG} is defined in Deliverable 1.4 as a document that will contain a set of recommendations to enable multilayer cyber security features in IIoT, including an architecture and methodology to provision signed certificates with Hardware Security Module (HSM) and trusted material to devices with or without a Trusted Platform Module (TPM). The recommendations follow the security standard 62443 to distribute digital identities using security hardware.[14]

After a revision of all requirements in this second deliverable of the requirements analysis and functional specification task, the following requirements are set up for the i4Q^{CSG} (Table 3).

ID	Type of Requirement	Requirement Title	Requirement definition (related to the i4Q Solution)	Requirement coming from connected Solution
05IKERr1	Guidelines req	Provide trust mechanisms	The cybersecurity guidelines shall provide trust mechanism based on IEC 62443 standard.	5-i4Q_CSG
05IKERr2	Guidelines req	Describe security mechanisms	The cybersecurity guidelines should describe security mechanism both for an IACS (Industrial Automation & Control Systems) topology and for indidivual components.	5-i4Q_CSG

Table 3. Requirements mapped to i4Q^{CSG}

4.6 i4Q^{SH} IIoT Security Handler

i4Q^{SH} is defined in Deliverable 1.4 as a piece of software that distributes trust using x509 certificates and asymmetric cryptography. Once the trust is distributed, every module will have a digital identity that will use to provide security between different endpoints using asymmetric cryptography considering different trust policies, adjusting security and safety policies at different levels. i4QSH also exposes cryptography operations supported by an HSM, which other modules can exploit to ensure the trustability and privacy of data.[15]

The figures below provide the following information:

(1) Figure 38 shows the FSD of i4Q^{SH} with its inputs and outputs as well as the functions and sub-functions. Interfaces to other i4Q Solutions are also included. This provides a structural overview of the functionalities of the solution.

[14] This description is mainly taken from i4Q Deliverable 1.4 (Nowak-Meitinger and Jochem, 2021, p. 72).
[15] This description is mainly taken from i4Q Deliverable 1.4 (Nowak-Meitinger and Jochem, 2021, p. 73f).

(2) Figure 39 shows the mapped requirements to the solution functions. All pilot and solution requirements are included. This provides an overview of the current level of completeness of the requirements analysis and functional specification of the solution.

(3) The mapping results are analysed and discussed according to the evaluation criteria described in Section 2.3.7:

- ***Completeness*** – The mapping diagram shows that 5 out of 12 sub-functions (lowest level) have mapped requirements, which corresponds to a completeness value of 0.42. The "blank spots" where no requirements are mapped in the diagrams could carry the risk of (still) unknown requirements not being fulfilled, while at the same time giving developers room to act because no specific requirements are given in the use case scenarios.
- ***Precision*** – The mapping diagram shows that 3 out of 8 requirements have been mapped to sub-functions (lowest level), which corresponds to a precision value of 0.38. It should be checked, if further specification by users (pilots) and solution providers is needed to be more precise in the current requirements descriptions and their mapping.
- ***Interface specification*** – In the FSD there are "any i4Q Solution" identified as interfaces. Since 6-i4Q_SH should be connected to several i4Q Solutions these interfaces should be defined and clarified in the upcoming work packages 3 to 5 to ensure the interoperability.
- ***Req Origin*** – The requirements have been defined by 4 different stakeholders. To ensure that this solution is created in such a way that many other i4Q Solutions can use it, collaboration with all affected stakeholders should be considered.

The unprecise mapping of some of the requirements may be due to the fact that $i4Q^{SH}$ is a core solution that will be used by many others or by the Pilots as a whole. Additionally, the lowest level sub-blocks are in some cases very specific, and it only makes sense to map a higher-level block (even if not the solution as a whole). As it will have interactions with almost every other Solution in the next steps, a strong collaboration with the rest of BUILD WPs will be needed.

The procedure for creating the diagrams and the purpose are described in Section 2. Further descriptions of the solution can be found in Deliverable 2.7 as well as in Deliverable 1.4. The functions of the solution are described in detail in Deliverable 2.5.

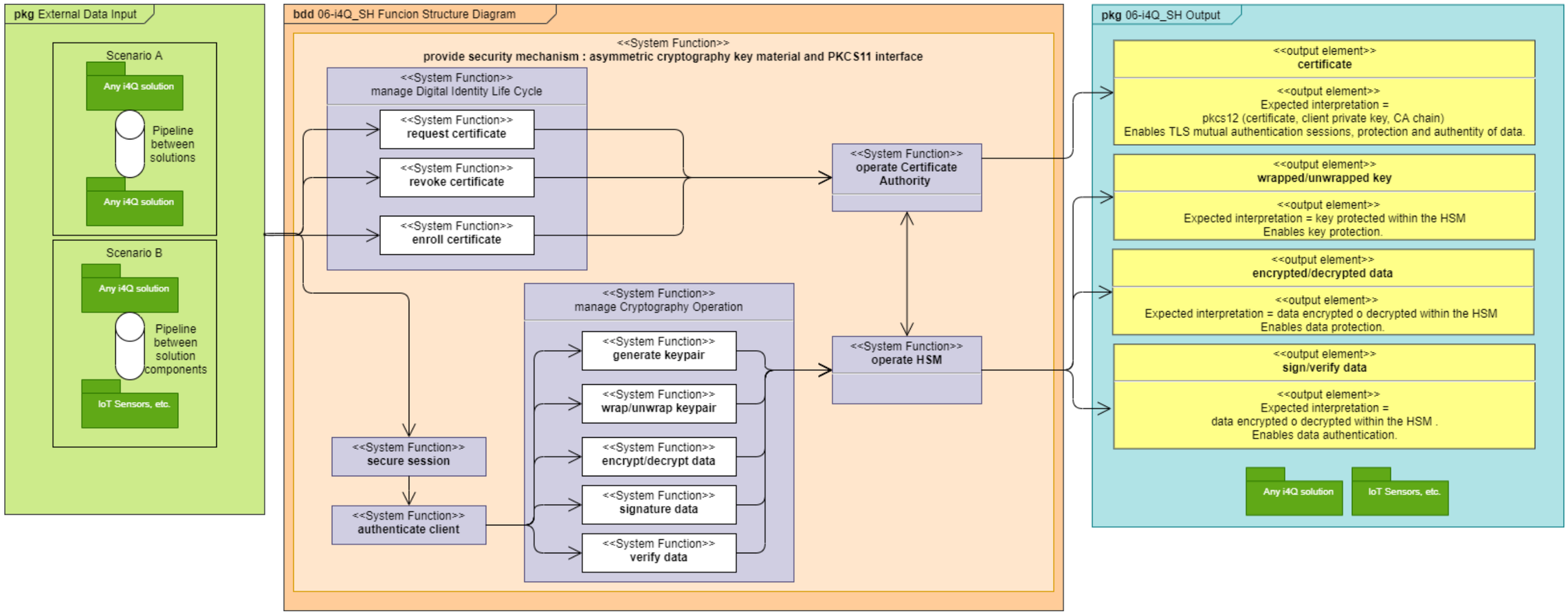


Figure 38. i4Q^{SH} Function Structure Diagram (FSD)

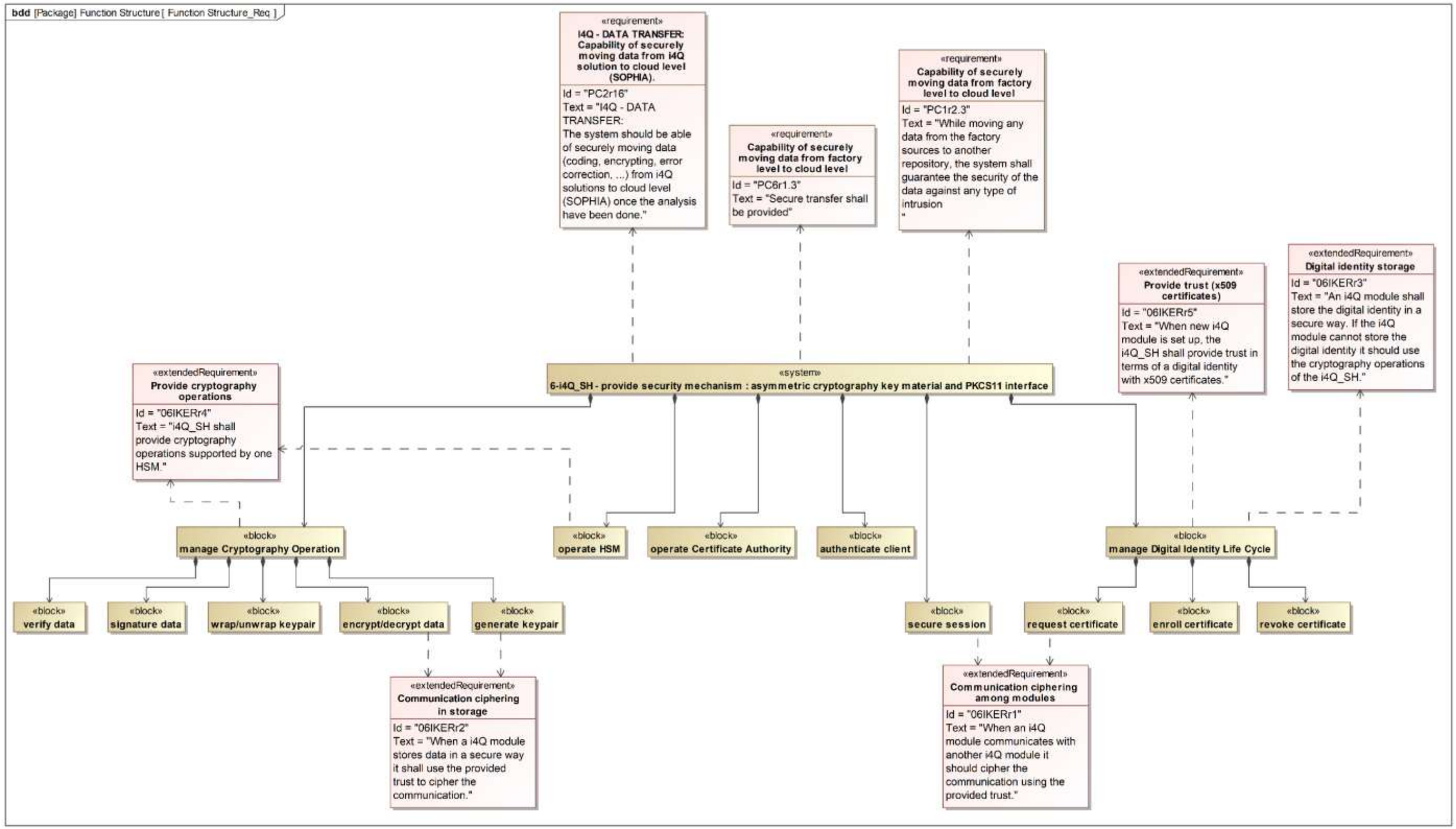


Figure 39. i4Q^{SH} Requirements Mapping and Functional Specification

4.7 i4Q[DRG] Guidelines for building Data Repositories for Industry 4.0

i4Q[DRG] is defined in Deliverable 1.4 as a document that will contain guidelines to build data repositories in the industry 4.0 paradigm. The guideline is targeted to the system administration and development team of a manufacturing company intends to transition to such paradigm. The goal is to pass on the knowledge gathered on the build process of one of these repositories, giving advice to follow and presenting errors to avoid. Possible topics to be treated are project planning, architecture design, data processing (Normalization, encryption, etc.), performance, software alternatives, implementation details, integration with other technologies and transition from legacy systems. Any other relevant topics will be identified and treated in order to get the most complete reference possible.[16]

After a revision of all requirements in this second deliverable of the requirements analysis and functional specification task, the following requirements are set up for the i4Q[DRG] (Table 4).

ID	Type of Requirement	Requirement Title	Requirement definition (related to the i4Q Solution)	Requirement coming from connected Solution
07ITIr1	Guidelines req, Usability and Quality req	Describe repository use	A technical document describing the whole process of the repository creation shall be provided. It will be as intuitive and complete as possible, to serve as a guide for future implementations. It will take some input from i4Q_DR.	7-i4Q_DRG, 8-i4Q_DR
PC3r3.3	Functional /Tehnical req.	All data generated by new modules shall be adhering to provided ontology MPFQ (developed in H2020-QU4LITY) Evaluate and embed MPFQ model into i4Q to harmonize the data.	The existing MPFQ (Material-Process-Functions-Quality) Model should be evaluated and embedded, and the data should be mapped (harmonized) on it	7-i4Q_DRG

[16] This description is mainly taken from i4Q Deliverable 1.4 (Nowak-Meitinger and Jochem, 2021, p. 76).

Table 4. Requirements mapped to i4Q^{DRG}

4.8 i4Q^{DR} Data Repository

i4Q^{DR} is defined in Deliverable 1.4 as a system or repository that will oversee receiving, storing, and serving the data in a proper way to the other components in the architecture. It will provide the proper tools for administrators to consult and transform the information contained inside it, as well as ways for data scientist to use this data in their experimentation. These tools will be provided through a suitable user interface. The solution will also oversee the data protection, serving as a secure system for the information by means of encryption, both in flight and at rest. The result of this task will be an efficient repository, ready to provide its service to the rest of the components present in the system.[17]

The figures below provide the following information:

(1) Figure 40 shows the FSD of i4Q^{DR} with its input and outputs as well as the functions and sub-functions. Interfaces to other i4Q Solutions are also included. This provides a structural overview of the functionalities of the solution.
(2) Figure 41 shows the mapped requirements to the solution functions. All pilot and solution requirements are included. This provides an overview of the current level of completeness of the requirements analysis and functional specification of the solution. The complete requirements mapping to i4Q^{DR} is also included in the matrix in **Appendix II**.
(3) The mapping results are analysed and discussed according to the evaluation criteria described in Section 2.3.7:
    - ***Completeness*** – The mapping diagram shows that 7 out of 18 sub-functions (lowest level) have mapped requirements, which corresponds to a completeness value of 0.39. The "blank spots" where no requirements are mapped in the diagrams could carry the risk of (still) unknown requirements not being fulfilled, while at the same time giving developers room to act because no specific requirements are given in the use case scenarios. There are also two “hot spots” in the diagram, the sub-functions “save image blob” and “save structured data” have many mapped requirements to satisfy as they define the main functions of the solution.
    - ***Precision*** – The mapping diagram shows that 31 out of 39 requirements have been mapped to sub-functions (lowest level), which corresponds to a precision value of 0.79. It should be checked, if further specification by users (pilots) and solution providers is needed in the current requirements descriptions and their mapping.

[17] This description is mainly taken from i4Q Deliverable 1.4 (Nowak-Meitinger and Jochem, 2021, p. 79f).

- *Interface specification* – In the FSD 4 input and 8 output interfaces are identified and defined. Since 8-i4Q_DR will be connected to several i4Q Solutions, these interfaces should be clarified in the WPs 3 to 5 to ensure the interoperability.
- *Req Origin* – The requirements have been defined by 12 different stakeholders. To ensure that this solution is created in such a way that many other i4Q Solutions can use it, collaboration with all affected stakeholders should be considered.

Regarding the completeness ratio, there are several functions and subfunctions that are not explicitly mapped to requirements, but they are assumed to be needed, in order to properly support other explicitly mapped functions. For instance, this is the case of the subfunctions of the "config access control" function. In order to have a properly working access control mechanism, some regular mechanisms to configure it are actually required. This is also the case of the subfunctions of "access control" and "manage data repository". Similarly, the subfunctions "delete structured data" and "delete image blob" are not explicitly required but they are also assumed to exist. Regarding the Interface specification and Req Origin criteria, the interfaces with other solutions can be easily extended to reach additional solutions as long as the new interoperability requirements are related to regular data storage needs. A more careful and detailed analysis would be required to add more specific interoperability requirements.

The procedure for creating the diagrams and the purpose are described in Section 2. Further descriptions of the solution can be found in Deliverable 2.7 as well as in Deliverable 1.4. The functions of the solution are described in detail in Deliverable 2.5.

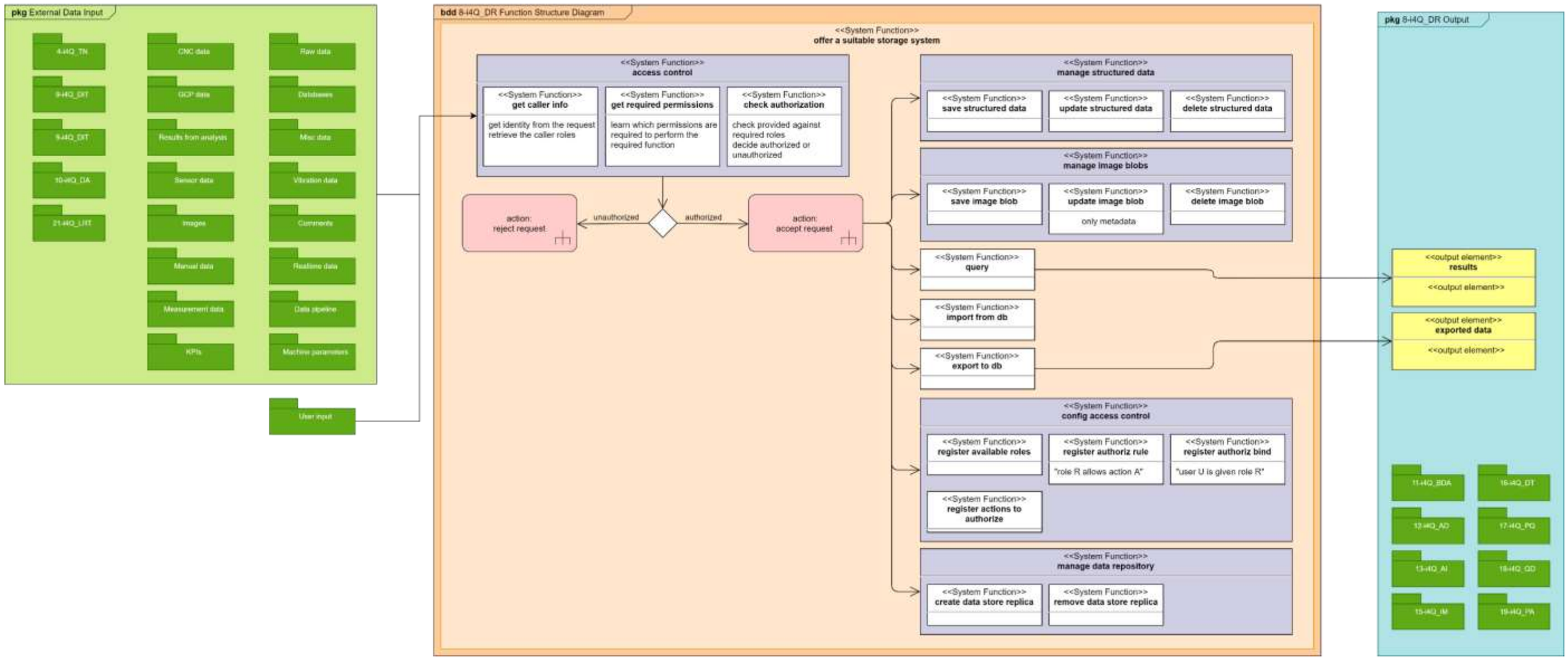


Figure 40. i4Q[DR] Function Structure Diagram (FSD)

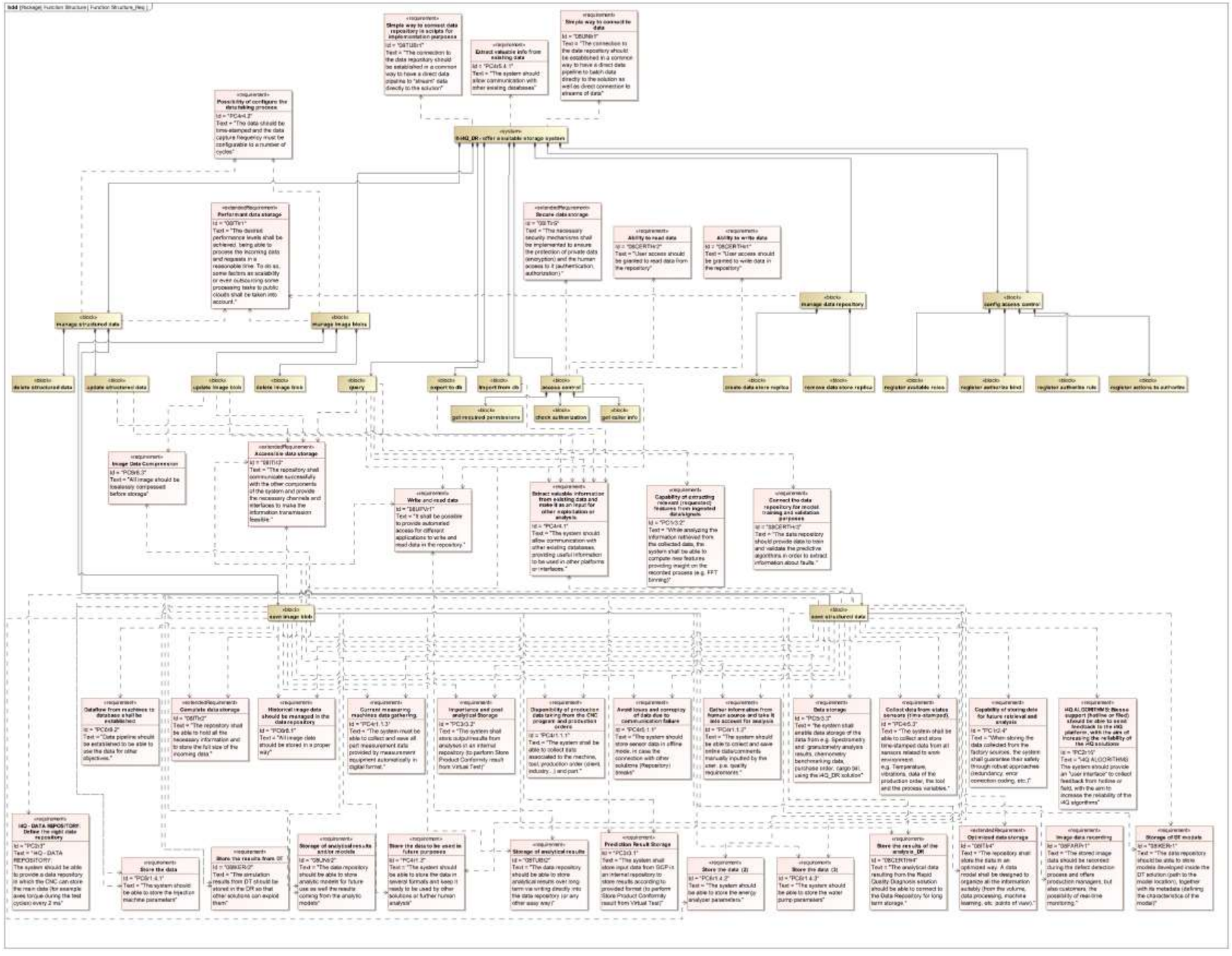


Figure 41. i4Q^{DR} Requirements Mapping and Functional Specification

4.9 i4Q^{DIT} Data Integration and Transformation Services

i4Q^{DIT} is defined in Deliverable 1.4 as a system whose main functions are a) to clean and process the input data so that they will be ready for further analysis/ modelling and b) to provide a framework for fusing the different types of data derived from the various sensors involved in the pilots. It will interact with other solutions to receive data, as well as provide them in other formats. This solution, that will be a distributed server-based platform, will contain other microservices for conducting all the preparation and decision-making actions; Some of these microservices are: data cleaning, sensor signals filtering, feature extraction, early fusion of heterogeneous data.[18]

The figures below provide the following information:

(1) Figure 42 shows the FSD of i4Q^{DIT} with its input and outputs as well as the functions and sub-functions. Interfaces to other i4Q Solutions are also included. This provides a structural overview of the functionalities of the solution.
(2) Figure 43 shows the mapped requirements to the solution functions. All pilot and solution requirements are included. This provides an overview of the current level of completeness of the requirements analysis and functional specification of the solution.
(3) The mapping results are analysed and discussed according to the evaluation criteria described in Section 2.3.7:
    - *Completeness* – The mapping diagram shows that 7 out of 9 sub-functions (lowest level) have mapped requirements, which corresponds to a completeness value of 0.78. The "blank spots" where no requirements are mapped in the diagrams could carry the risk of (still) unknown requirements not being fulfilled, while at the same time giving developers room to act because no specific requirements are given in the use case scenarios. There are also two “hot spots” in the diagram, the sub-functions “import sensor data” and “harmonize data” have many mapped requirements to satisfy as they define the main functions of the solution.
    - *Precision* – The mapping diagram shows that 27 out of 32 requirements have been mapped to sub-functions (lowest level), which corresponds to a precision value of 0.84. It should be checked, if further specification by users (pilots) and solution providers is needed in the current requirements descriptions and their mapping.
    - *Interface specification* – In the FSD 1 input and output interface to the 8-i4Q_DR is identified and defined. This interface should be clarified in the upcoming work packages 3 to 5 to ensure the interoperability.

[18] This description is mainly taken from i4Q Deliverable 1.4 (Nowak-Meitinger and Jochem, 2021, p. 86).

- *Req Origin* – The requirements have been defined by 10 different stakeholders. To ensure that this solution is created in such a way that many other i4Q Solutions can use it, collaboration with all affected stakeholders should be considered.

The existence of "blank spots" in the diagram is due to the fact that there is no clear definition of the data format deriving from each requirement. As a result, higher level functions had to be mapped to some requirements to cover those specific low-level sub functions. Furthermore, 9_i4Q_DIT is the main solution responsible for pre-processing data and creating integrated datasets that are going to be used by other solutions meaning that the interface between 8_i4Q_DR and 9_i4Q_DIT might not be unique and other interfaces between the solution and the corresponding partner need to be created.

The procedure for creating the diagrams and the purpose are described in Section 2. Further descriptions of the solution can be found in Deliverable 2.7 as well as in Deliverable 1.4. The functions of the solution are described in detail in Deliverable 2.5.

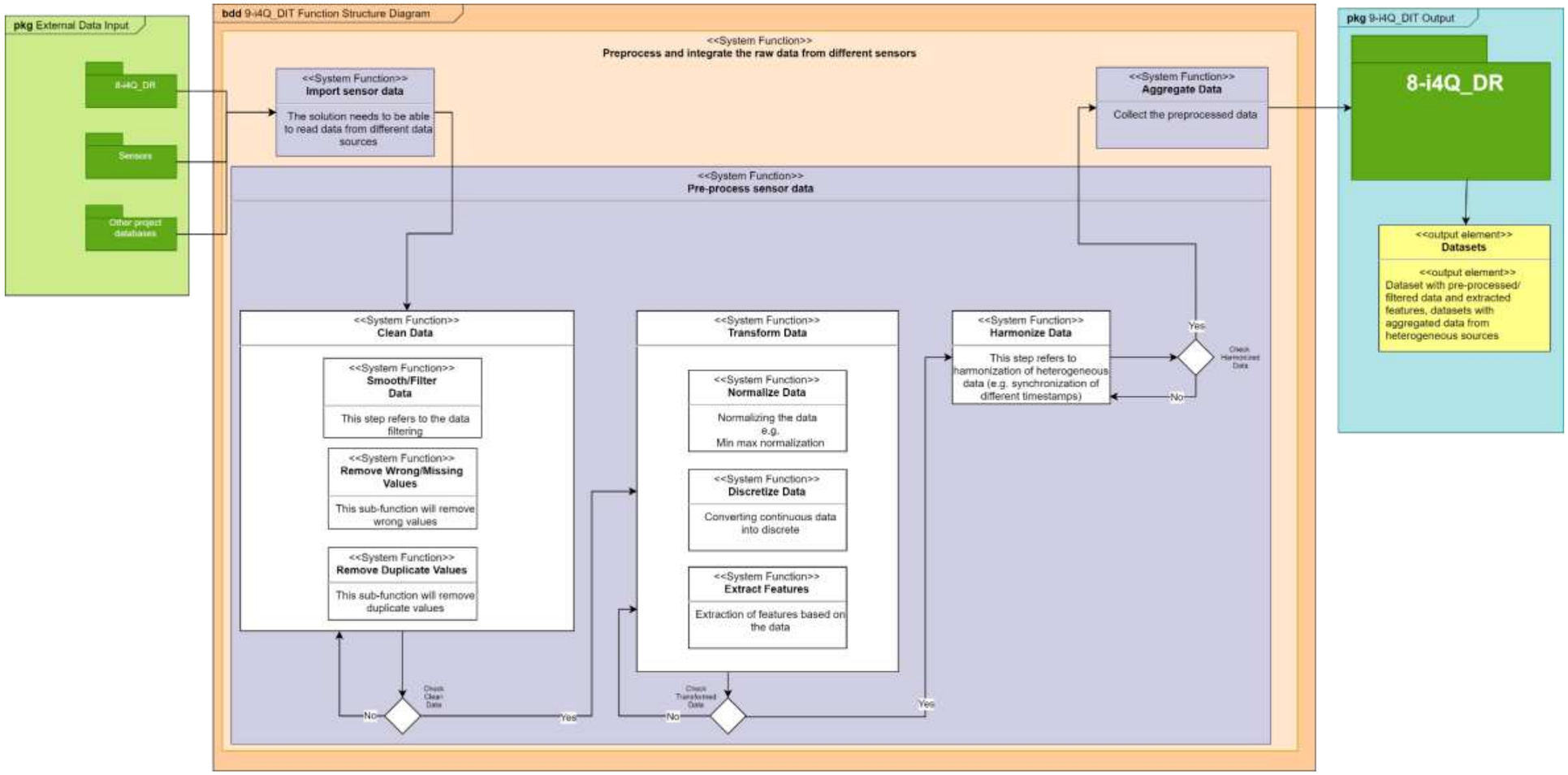


Figure 42. i4Q^{DIT} Function Structure Diagram (FSD)

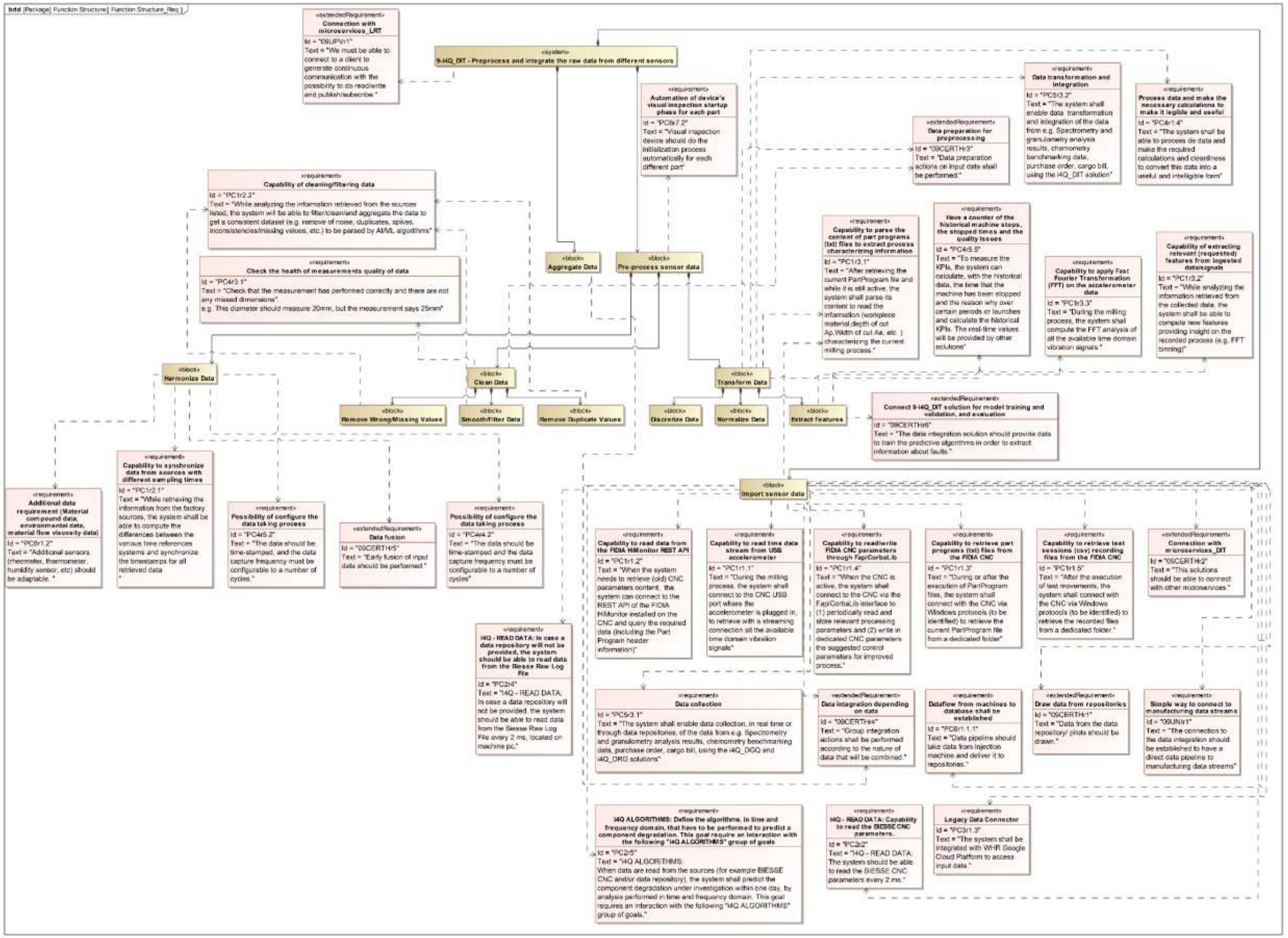


Figure 43. i4Q^{DIT} Requirements Mapping and Functional Specification

4.10 i4Q^{DA} Services for Data Analytics

i4Q^{DA} is defined in Deliverable 1.4 as a system whose main functions are the provision of Data Analytics services, supported by the integration of several state-of-the-art tools, methods, and libraries, ranging from Big Data Processing and Analytics to Machine Learning, Data Mining and Deep Learning. The services will be provisioned through two main channels: i) Open Application Programming Interfaces (APIs) like RESTFul-based, pub-sub, socket-based for the collection of the necessary data to execute the selected services and for the provision of results coming from the Data Analytics services, or ii) Deployment bundles with the necessary tools, methods, and libraries to deploy and run the selected services on premises or on cloud environments.[19]

The figures below provide the following information:

[19] This description is mainly taken from i4Q Deliverable 1.4 (Nowak-Meitinger and Jochem, 2021, p. 95).

(1) Figure 44 shows the FSD of i4Q[DA] with its input and outputs as well as the functions and sub-functions. Interfaces to other i4Q Solutions are also included. This provides a structural overview of the functionalities of the solution.
(2) Figure 45 shows the mapped requirements to the solution functions. All pilot and solution requirements are included. This provides an overview of the current level of completeness of the requirements analysis and functional specification of the solution.
(3) The mapping results are analysed and discussed according to the evaluation criteria described in Section 2.3.7:
    - ***Completeness*** – The mapping diagram shows that 5 out of 13 sub-functions (lowest level) have mapped requirements, which corresponds to a completeness value of 0.38. The "blank spots" where no requirements are mapped in the diagrams could carry the risk of (still) unknown requirements not being fulfilled, while at the same time giving developers room to act because no specific requirements are given in the use case scenarios. The sub-functions "execute models" and "deliver results" have three or more mapped requirements to satisfy.
    - ***Precision*** – The mapping diagram shows that 12 out of 20 requirements have been mapped to sub-functions (lowest level), which corresponds to a precision value of 0.60. It should be checked, if further specification by users (pilots) and solution providers is needed to be more precise in the current requirements descriptions and their mapping.
    - ***Interface specification*** – In the FSD 2 input and 3 output interfaces are identified and defined. These interfaces should be clarified in the upcoming work packages 3 to 5 to ensure the interoperability.
    - ***Req Origin*** – The requirements have been defined by 6 different stakeholders. To ensure that this solution is created in such a way that many other i4Q Solutions can use it, collaboration with all affected stakeholders should be considered.

The low precision of the mapping of the requirements is connected to the fact that some applications of the solutions are still unknown, which makes the mapping to low level functions of some requirements challenging. Additionally, the requirement descriptions contain more information than the solution can currently provide. For that reason, the requirements are mapped to the higher-level functions in the functional architecture which have to be addressed in the development process in WP4. Since we have to account for the collaboration with all stakeholders, more information will be available throughout the development of the project. This will improve both *Completeness* and *Precision*.

The procedure for creating the diagrams and the purpose are described in Section 2. Further descriptions of the solution can be found in Deliverable 2.7 as well as in Deliverable 1.4. The functions of the solution are described in detail in Deliverable 2.5.

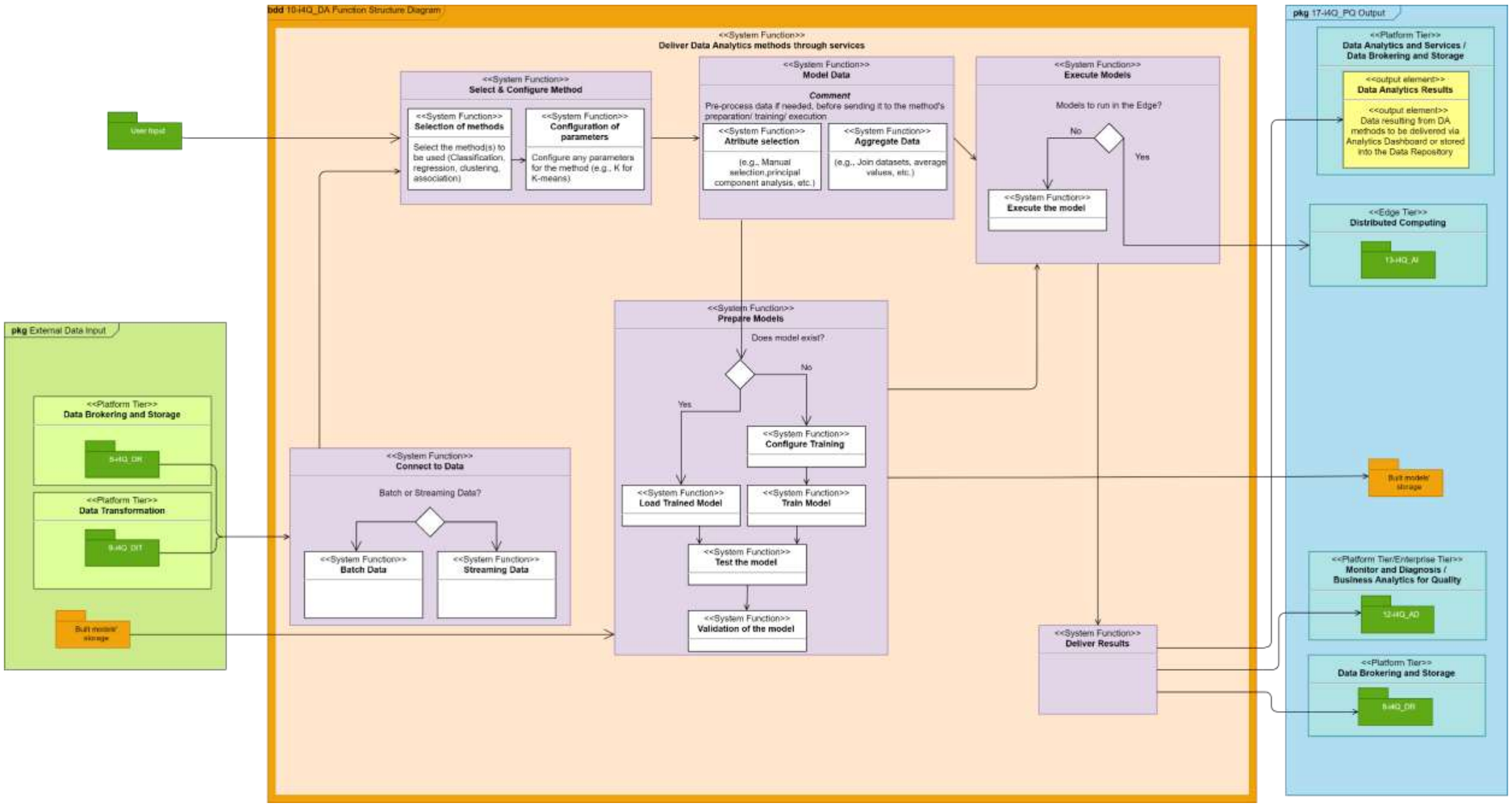


Figure 44. i4Q^{DA} Function Structure Diagram (FSD)

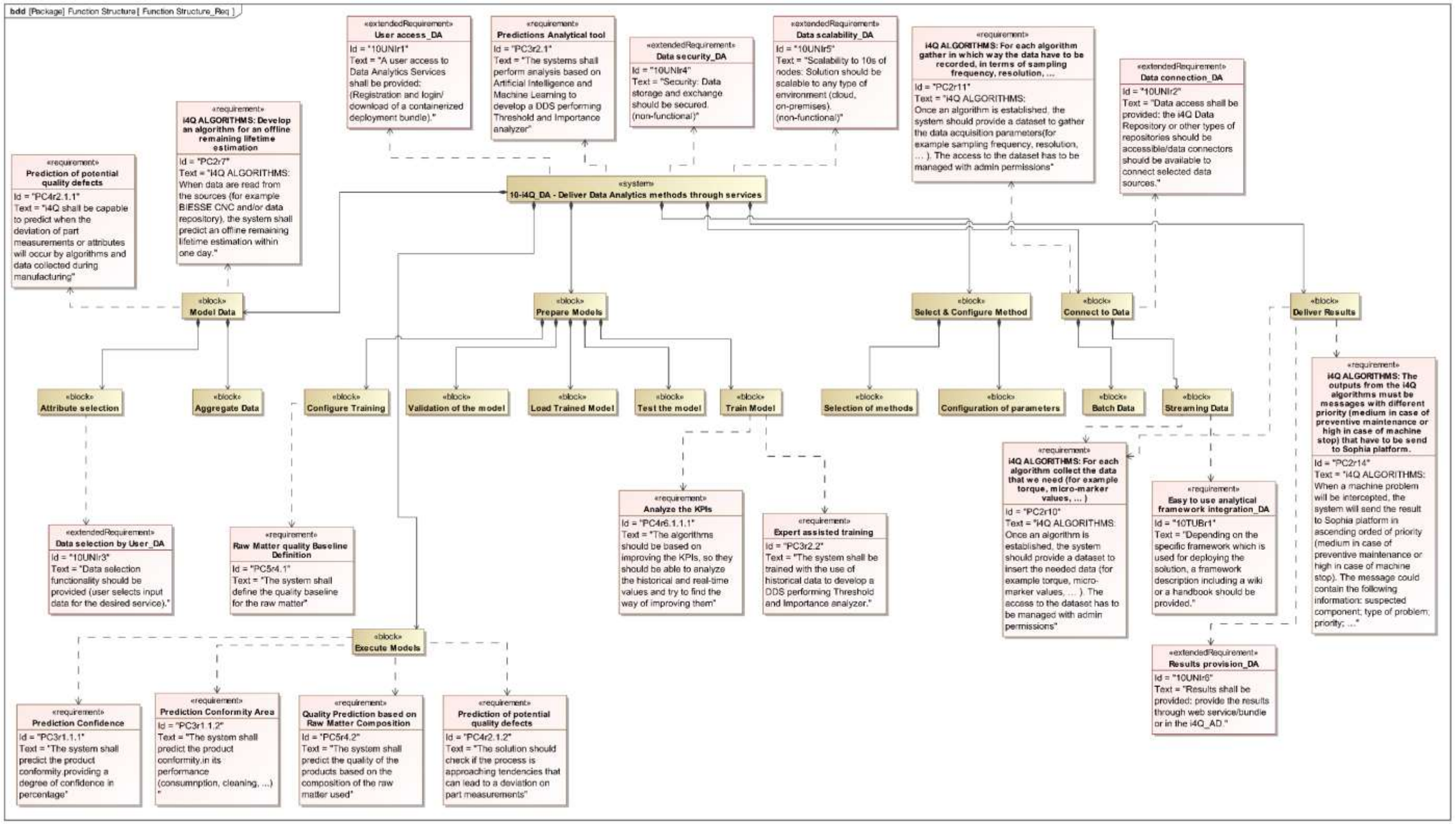


Figure 45. i4Q[DA] Requirements Mapping and Functional Specification

4.11 i4Q^{BDA} Big Data Analytics Suite

i4Q^{BDA} is defined in Deliverable 1.4 as a system whose main function is to deliver on-demand deployment bundles that are easily configurable, deployable, and executed. This Suite will be able to provide custom-built deployment bundles that can contain all the necessary tools, methods, libraries, and code to deploy and run the selected Data Analytics tasks in a panoply of environments, from centralized, distributed on-premises or Cloud. This solution will be supported by containerization technologies such as Kubernetes or Docker.[20]

The figures below provide the following information:

(1) Figure 46 shows the FSD of i4Q^{BDA} with its input and outputs as well as the functions and sub-functions. Interfaces to other i4Q Solutions are also included. This provides a structural overview of the functionalities of the solution.
(2) Figure 47 shows the mapped requirements to the solution functions. All pilot and solution requirements are included. This provides an overview of the current level of completeness of the requirements analysis and functional specification of the solution.
(3) The mapping results are analysed and discussed according to the evaluation criteria described in Section 2.3.7:
    - *Completeness* – The mapping diagram shows that 7 out of 11 sub-functions (lowest level) have mapped requirements, which corresponds to a completeness value of 0.64. The "blank spots" where no requirements are mapped in the diagrams could carry the risk of (still) unknown requirements not being fulfilled, while at the same time giving developers room to act because no specific requirements are given in the use case scenarios. The “hot spot” in the diagram is the function “select methods/techs”. This function has to satisfy many mapped requirements. As they could not be mapped to the lowest level sub-functions, they should be clarified precisely in the upcoming work packages.
    - *Precision* – The mapping diagram shows that 7 out of 29 requirements have been mapped to sub-functions (lowest level), which corresponds to a precision value of 0.24. It should be checked, if further specification by users (pilots) and solution providers is needed to be more precise in the current requirements descriptions and their mapping.
    - *Interface specification* – In the FSD 2 input and 0 output interfaces are identified and defined. Since the results of 11-i4Q_BDA should be used in other i4Q Solutions at least one output interface should be identified and added. All these interfaces should be then clarified in the upcoming work packages 3 to 5 to ensure the interoperability.

[20] This description is mainly taken from i4Q Deliverable 1.4 (Nowak-Meitinger and Jochem, 2021, p. 101).

- *Req Origin* – The requirements have been defined by 6 different stakeholders. To ensure that this solution is created in such a way that many other i4Q Solutions can use it, collaboration with all affected stakeholders should be considered.

The low precision of the mapping of the requirements is connected to the fact that some applications of the solutions are still unknown, which makes the mapping to low level functions of some requirements challenging. Additionally, the description of the requirement has more information than the solution is currently able to deliver. For that reason, the requirements are mapped to the higher-level functions in the functional architecture which have to be addressed in the development process in WP4. Since we have to account for the collaboration with all stakeholders, more information will be available throughout the development of the project. This will improve both *Completeness* and *Precision*.

The procedure for creating the diagrams and the purpose are described in Section 2. Further descriptions of the solution can be found in Deliverable 2.7 as well as in Deliverable 1.4. The functions of the solution are described in detail in Deliverable 2.5.

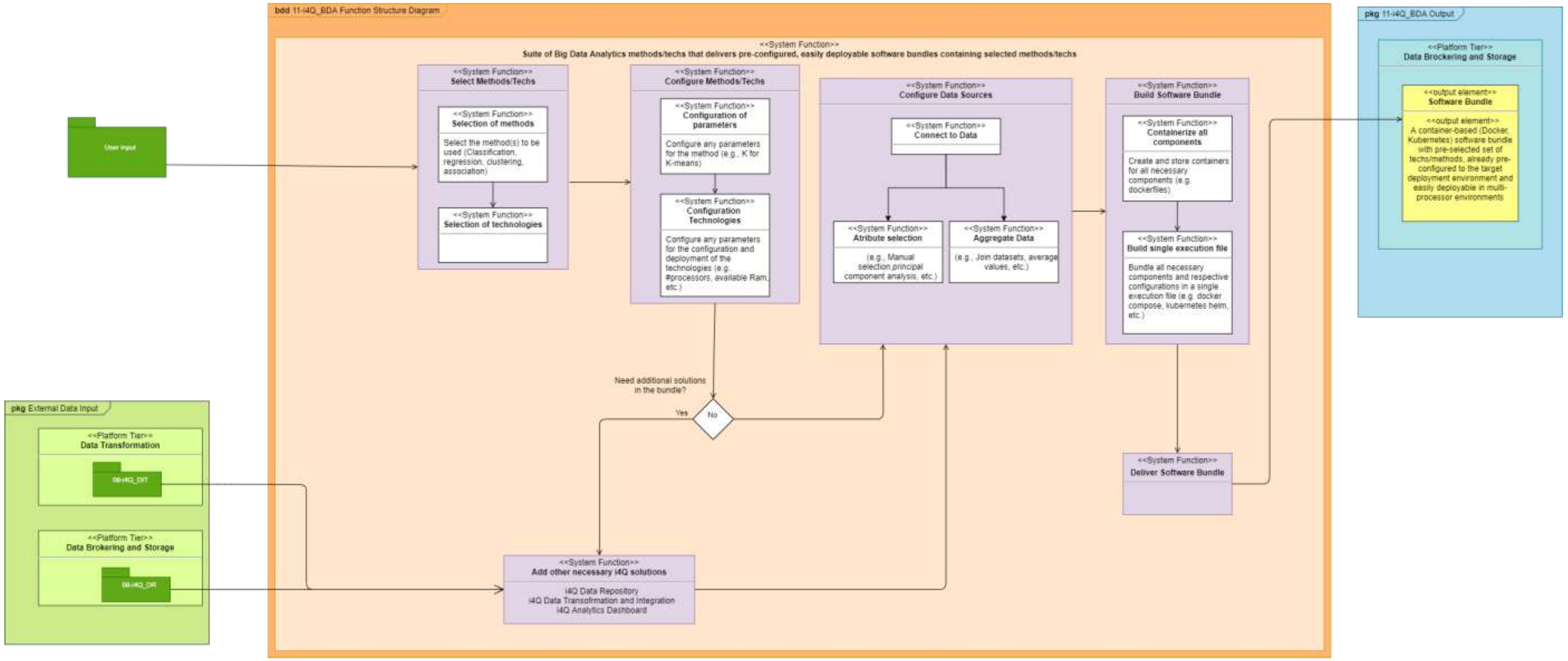


Figure 46. i4Q^{BDA} Function Structure Diagram (FSD)

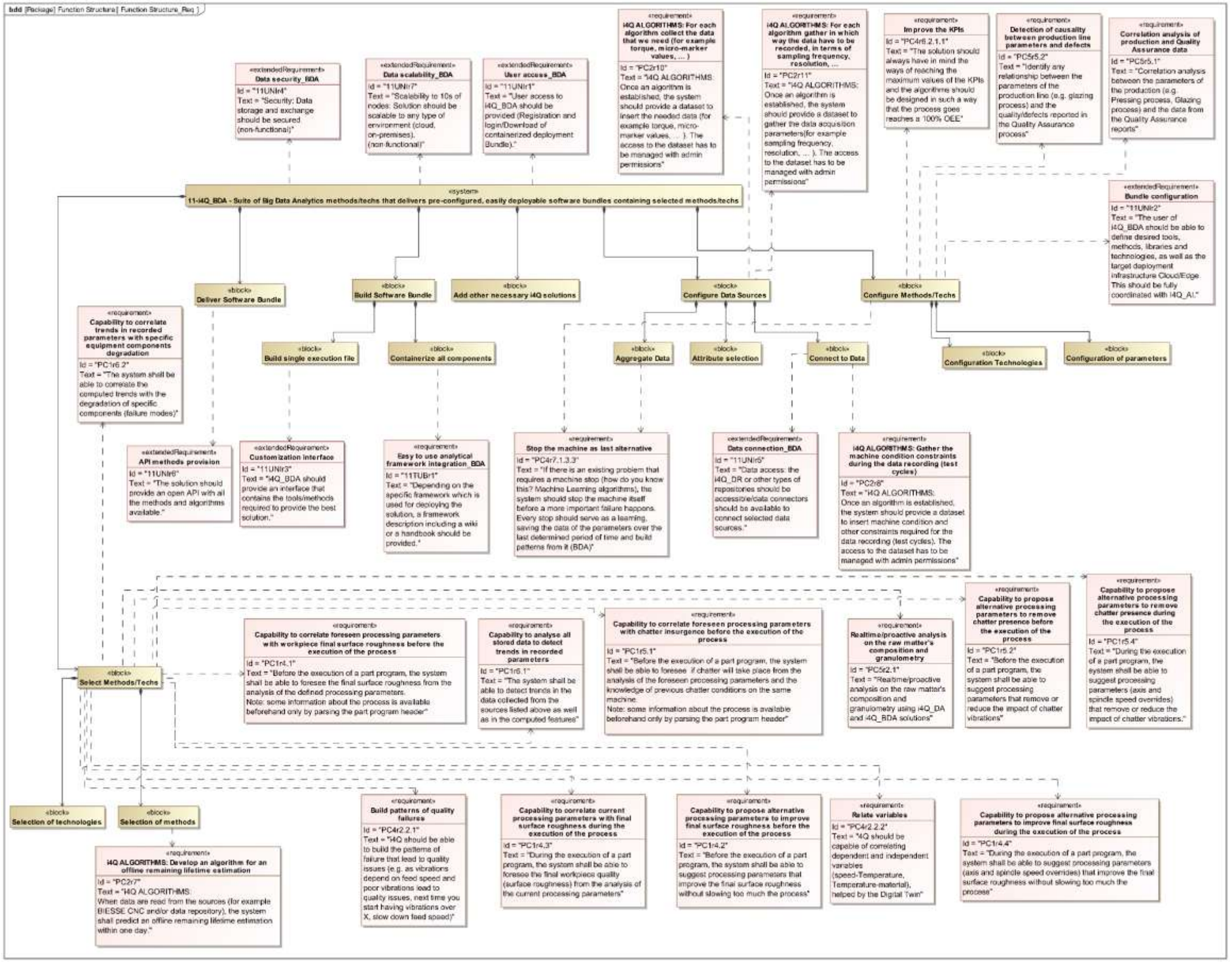


Figure 47. i4Q^{BDA} Requirements Mapping and Functional Specification

4.12 i4Q^{AD} Analytics Dashboard

i4Q^{AD} is defined in Deliverable 1.4 as a system whose main function is to provide visual analytics tools and methods to the i4Q project. The i4Q Analytics Dashboard can be used via a Web Application or through the provision of a deployment bundle that can be deployed on premises or on the cloud, and will be based on state-of-the-art visual analytics tools, such as Apache Superset, Grafana or Jupyter Notebooks.[21]

The figures below provide the following information:

(1) Figure 48 shows the FSD of i4Q^{AD} with its input and outputs as well as the functions and sub-functions. Interfaces to other i4Q Solutions are also included. This provides a structural overview of the functionalities of the solution.
(2) Figure 49 shows the mapped requirements to the solution functions. All pilot and solution requirements are included. This provides an overview of the current level of completeness of the requirements analysis and functional specification of the solution.
(3) The mapping results are analysed and discussed according to the evaluation criteria described in Section 2.3.7:
    - ***Completeness*** – The mapping diagram shows that 6 out of 10 sub-functions (lowest level) have mapped requirements, which corresponds to a completeness value of 0.60. The "blank spots" where no requirements are mapped in the diagrams could carry the risk of (still) unknown requirements not being fulfilled, while at the same time giving developers room to act because no specific requirements are given in the use case scenarios.
    - ***Precision*** – The mapping diagram shows that 14 out of 26 requirements have been mapped to sub-functions (lowest level), which corresponds to a precision value of 0.54. It should be checked, if further specification by users (pilots) and solution providers is needed to be more precise in the current requirements descriptions and their mapping.
    - ***Interface specification*** – In the FSD 7 input and 0 output interfaces are identified and defined. This solution is receiving mainly data from other solutions to visualize them. The solution interfaces should be then clarified in the upcoming work packages 3 to 5 to ensure the interoperability.
    - ***Req Origin*** – The requirements have been defined by 8 different stakeholders including all pilots. To ensure that this solution is created in such a way that many other i4Q Solutions can use it, collaboration with all affected stakeholders should be considered.

The average precision of the mapping of the requirements is connected to the fact that some applications of the solutions are still unknown, which makes the mapping to low level functions of some requirements challenging. Additionally, the description of the requirement has more

[21] This description is mainly taken from i4Q Deliverable 1.4 (Nowak-Meitinger and Jochem, 2021, p. 108).

information than the solution is currently able to deliver. For that reason, the requirements are mapped to the higher-level functions in the functional architecture which have to be addressed in the development process in WP4. Since we have to account for the collaboration with all stakeholders, more information will be available throughout the development of the project. This will improve both *Completeness* and *Precision*.

The procedure for creating the diagrams and the purpose are described in Section 2. Further descriptions of the solution can be found in Deliverable 2.7 as well as in Deliverable 1.4. The functions of the solution are described in detail in Deliverable 2.5.

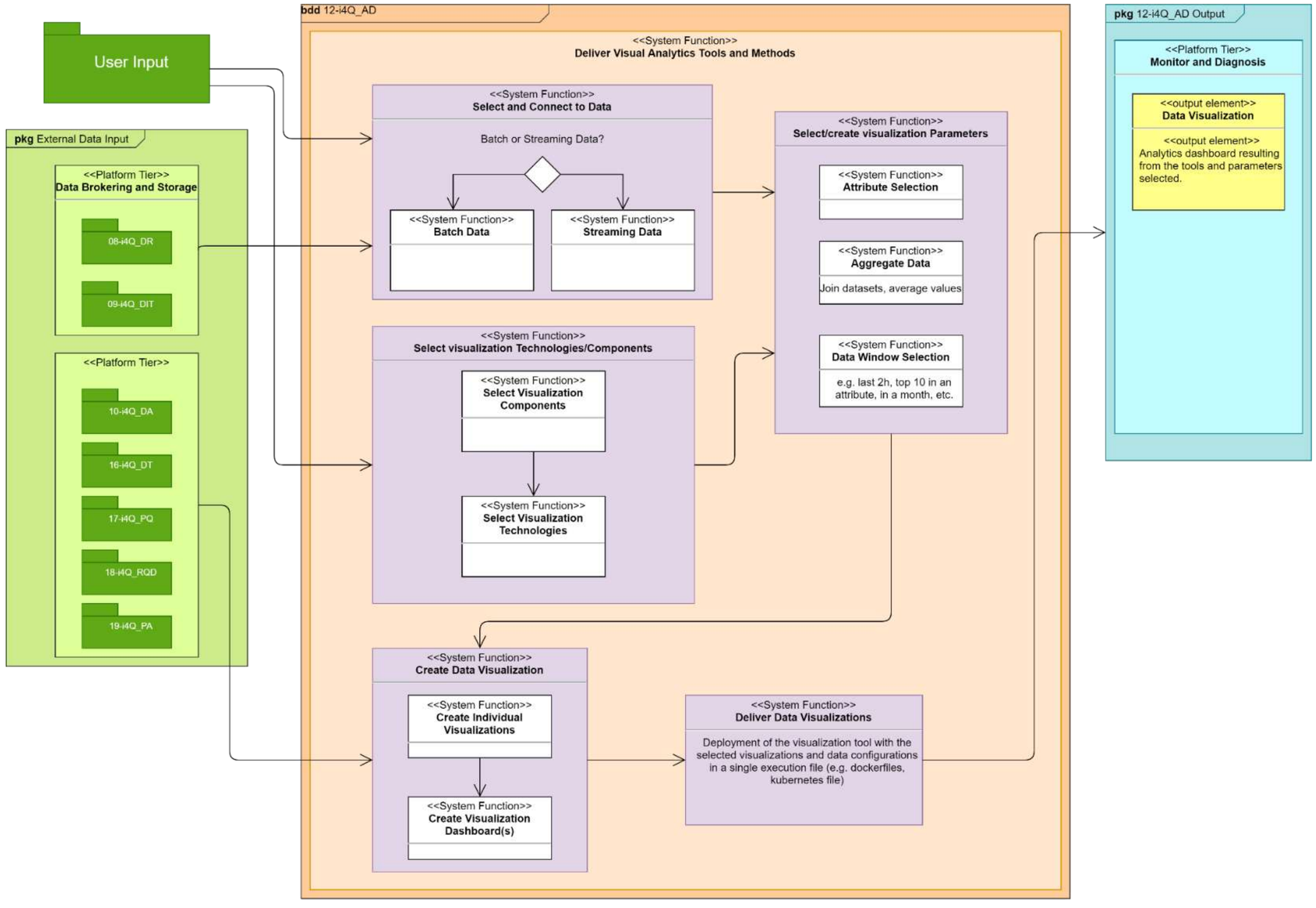


Figure 48. i4Q[AD] Function Structure Diagram (FSD)

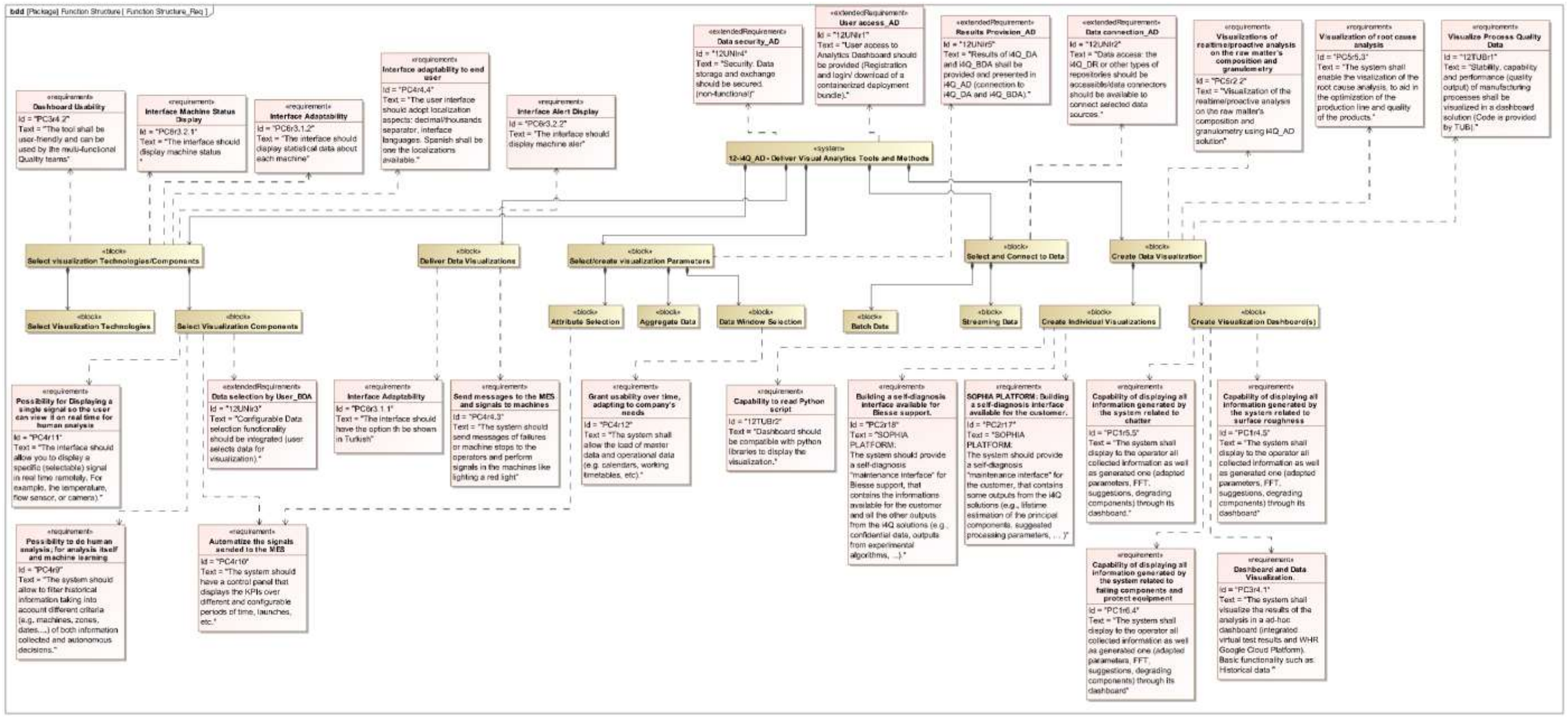


Figure 49. i4Q^{AD} Requirements Mapping and Functional Specification

4.13 i4Q^{AI} AI Models Distribution to the Edge

i4Q^{AI} is defined in Deliverable 1.4 as a system whose main function is the management of AI-based workloads in a hybrid cloud-edge manufacturing environment. A policy-based distribution mechanism shall be put in place to ease the process of model distribution to the correct edge nodes; aided by a discovery component keeping up-to-date information of system components. The AI model distribution shall be coordinated with the workload deployment mechanism to ensure that the right set of AI models is made available for the edge workload that uses them.[22]

The figures below provide the following information:

(1) Figure 50 shows the FSD of i4Q^{AI} with its input and outputs as well as the functions and sub-functions. Interfaces to other i4Q Solutions are also included. This provides a structural overview of the functionalities of the solution.
(2) Figure 51 shows the mapped requirements to the solution functions. All pilot and solution requirements are included. This provides an overview of the current level of completeness of the requirements analysis and functional specification of the solution.
(3) The mapping results are analysed and discussed according to the evaluation criteria described in Section 2.3.7:
    - ***Completeness*** – The mapping diagram shows that 2 out of 6 sub-functions (lowest level) have mapped requirements, which corresponds to a completeness value of 0.33. The "blank spots" where no requirements are mapped in the diagrams could carry the risk of (still) unknown requirements not being fulfilled, while at the same time giving developers room to act because no specific requirements are given in the use case scenarios.
    - ***Precision*** – The mapping diagram shows that 2 out of 3 requirements have been mapped to sub-functions (lowest level), which corresponds to a precision value of 0.67. It should be checked, if further specification by users (pilots) and solution providers is needed in the current requirements descriptions and their mapping.
    - ***Interface specification*** – In the FSD 2 input and 1 output interfaces are identified and defined, namely i4Q_EW and i4Q_IM. Since i4Q_AI should provide services to several i4Q Solutions such as i4Q_PQ, i4Q_QD and i4Q_DT, these interfaces should be added and clarified in the upcoming work packages 3 to 5 to ensure the interoperability.
    - ***Req Origin*** – The requirements have been defined by 3 different stakeholders. To ensure that this solution is created in such a way that many other i4Q Solutions can use it, collaboration with all affected stakeholders should be considered.

[22] This description is mainly taken from i4Q Deliverable 1.4 (Nowak-Meitinger and Jochem, 2021, p. 114).

The procedure for creating the diagrams and the purpose are described in Section 2. Further descriptions of the solution can be found in Deliverable 2.7 as well as in Deliverable 1.4. The functions of the solution are described in detail in Deliverable 2.5.

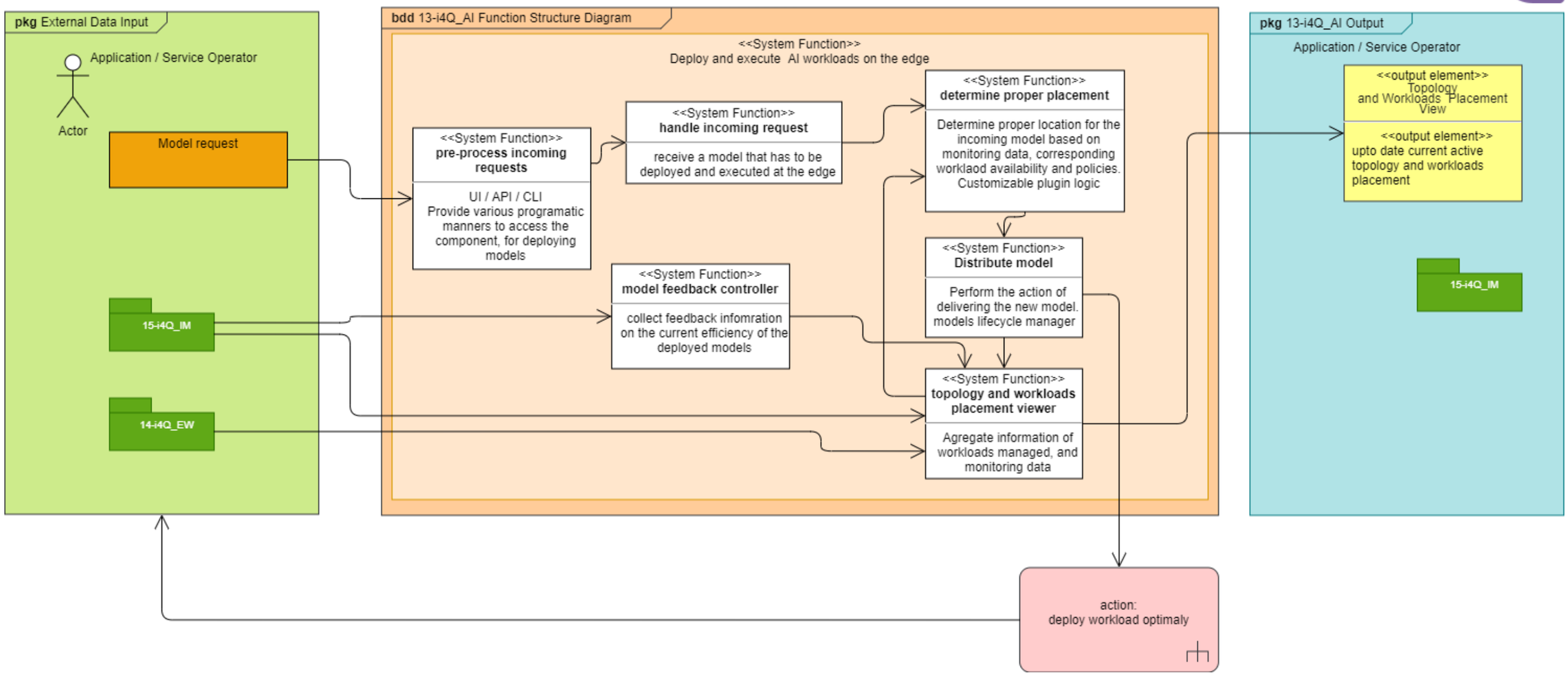


Figure 50. i4Q[AI] Function Structure Diagram (FSD)

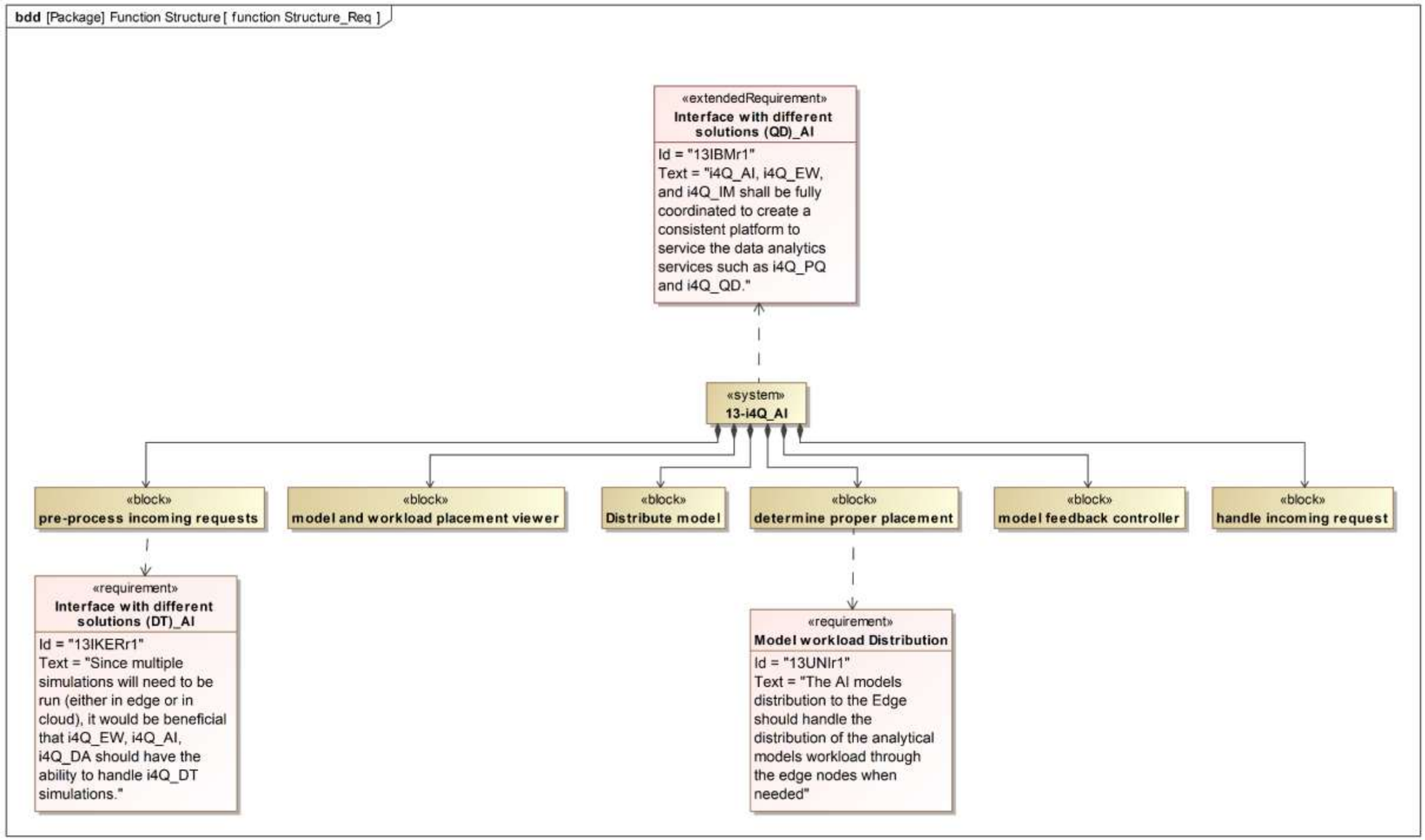


Figure 51. i4Q^{AI} Requirements Mapping and Functional Specification

4.14 i4Q^{EW} Workloads Placement and Deployment

i4Q^{EW} is defined in Deliverable 1.4 as a system whose main function is to enable workloads to be executed efficiently on the edge, including placement and deployment services. Target deployment environment may be very heterogeneous and dynamic; thus, deployment needs to take a variety of criteria into consideration. Interfaces and capabilities to run different workloads on different devices shall be pursued. Deployment shall be based on well-known orchestrators, such as Kubernetes.[23]

The figures below provide the following information:

(1) Figure 52 shows the FSD of i4Q^{EW} with its input and outputs as well as the functions and sub-functions. Interfaces to other i4Q Solutions are also included. This provides a structural overview of the functionalities of the solution.
(2) Figure 53 shows the mapped requirements to the solution functions. All pilot and solution requirements are included. This provides an overview of the current level of completeness of the requirements analysis and functional specification of the solution.
(3) The mapping results are analysed and discussed according to the evaluation criteria described in Section 2.3.7:
    - ***Completeness*** – The mapping diagram shows that 2 out of 6 sub-functions (lowest level) have mapped requirements, which corresponds to a completeness value of 0.33. The "blank spots" where no requirements are mapped in the diagrams could carry the risk of (still) unknown requirements not being fulfilled, while at the same time giving developers room to act because no specific requirements are given in the use case scenarios.
    - ***Precision*** – The mapping diagram shows that 3 out of 4 requirements have been mapped to sub-functions (lowest level), which corresponds to a precision value of 0.75. It should be checked, if further specification by users (pilots) and solution providers is needed in the current requirements descriptions and their mapping.
    - ***Interface specification*** – In the FSD 1 input and output interface is identified and defined, namely i4Q_IM. Since i4Q_EW should interact also with i4Q_AI and provide services to several i4Q Solutions such as i4Q_PQ, i4Q_QD, i4Q_LRT and i4Q_DT, all these interfaces should be added and clarified in the upcoming work packages 3 to 5 to ensure the interoperability.
    - ***Req Origin*** – The requirements have been defined by 4 different stakeholders. To ensure that this solution is created in such a way that many other i4Q Solutions can use it, collaboration with all affected stakeholders should be considered.

[23] This description is mainly taken from i4Q Deliverable 1.4 (Nowak-Meitinger and Jochem, 2021, p. 115).

The procedure for creating the diagrams and the purpose are described in Section 2. Further descriptions of the solution can be found in Deliverable 2.7 as well as in Deliverable 1.4. The functions of the solution are described in detail in Deliverable 2.5.

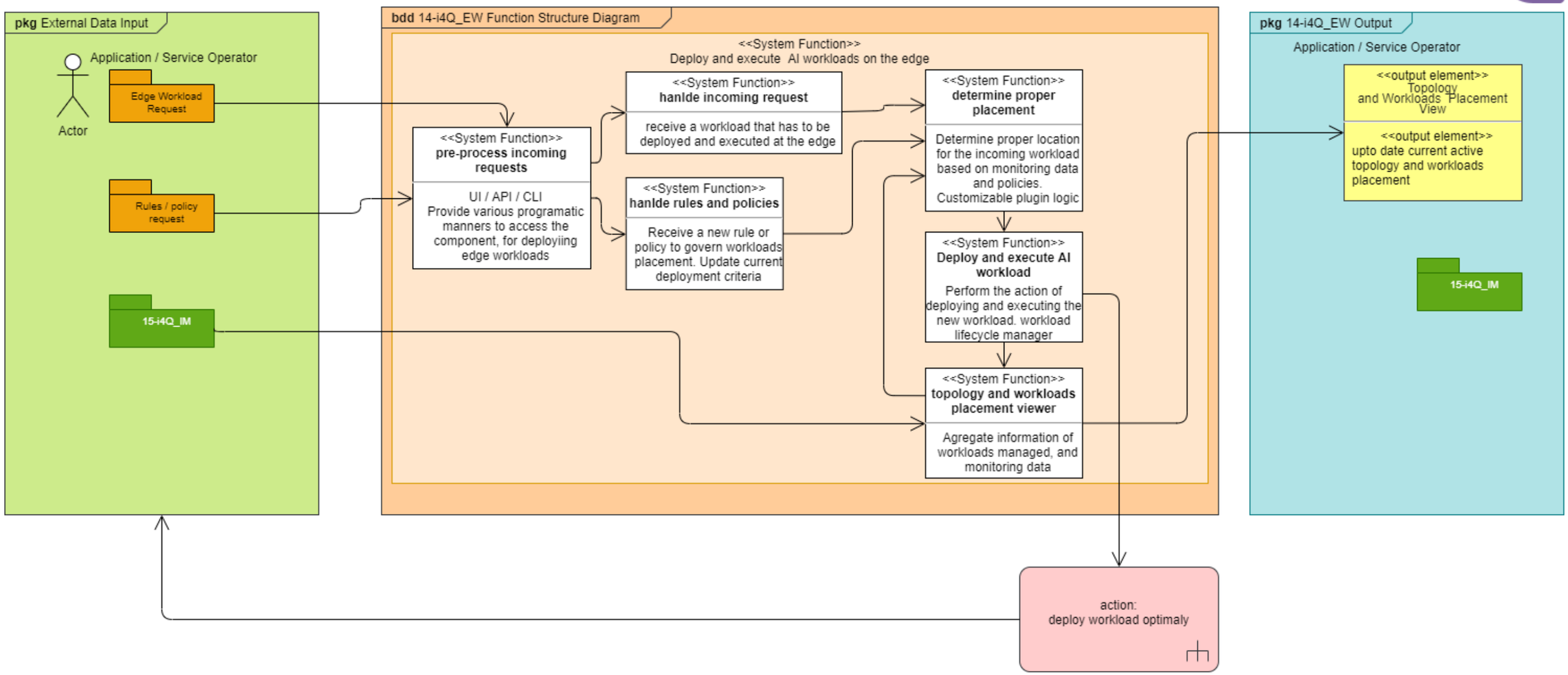


Figure 52. i4Q[EW] Function Structure Diagram (FSD)

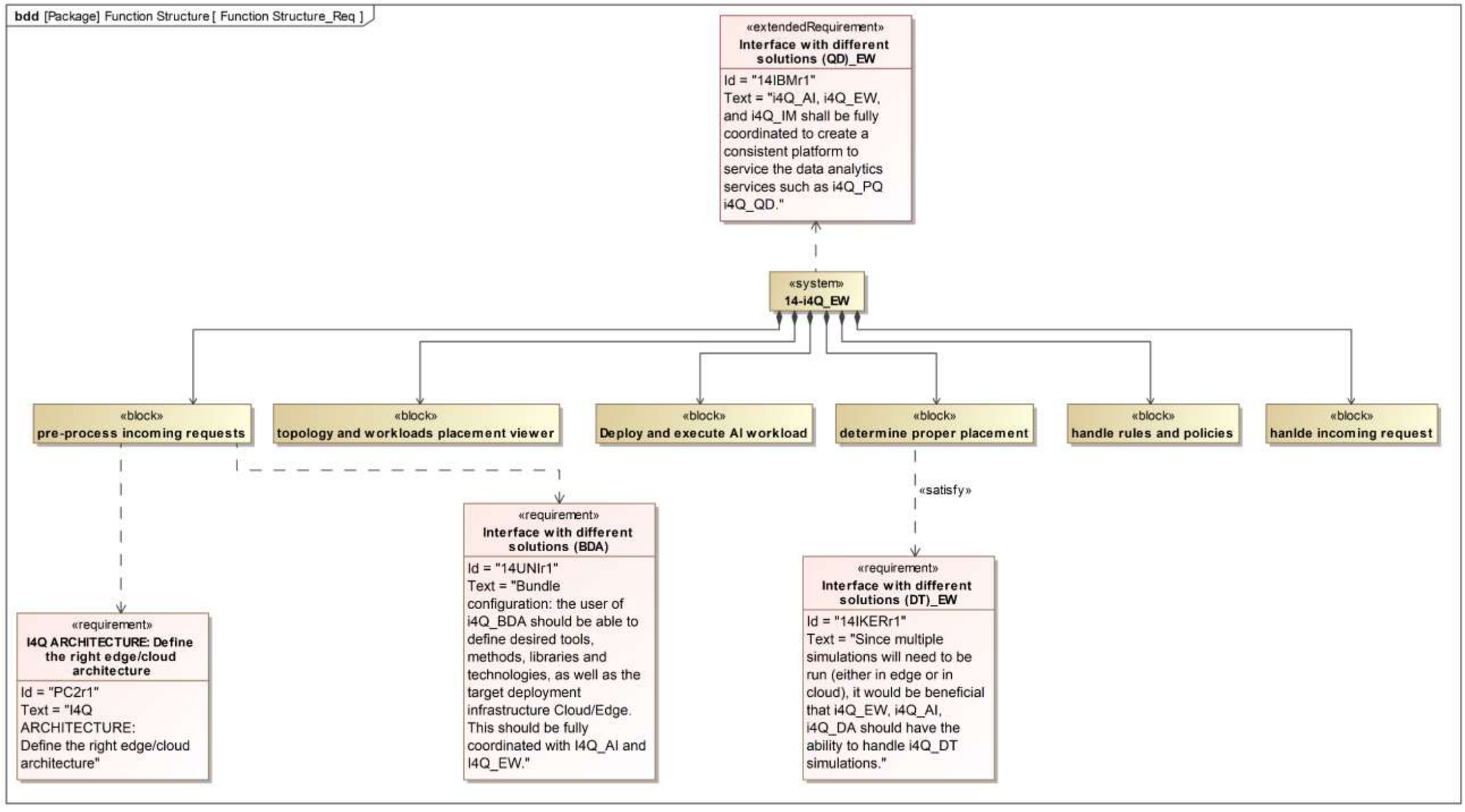


Figure 53. i4Q^{EW} Requirements Mapping and Functional Specification

4.15 i4Q^{IM} Infrastructure Monitoring

i4Q^{IM} is defined in Deliverable 1.4 as a system that will observe the manufacturing processes in order to predict possible failures and provide alerts or take corrective actions. In general, the solution will perform prediction of harmful events as well as detection of these failures, either in cooperation with other analytical solutions or as a standalone tool and will provide alerts in both cases.[24]

The figures below provide the following information:

(1) Figure 54 shows the FSD of i4Q^{IM} with its input and outputs as well as the functions and sub-functions. Interfaces to other i4Q Solutions are also included. This provides a structural overview of the functionalities of the solution.
(2) Figure 55 shows the mapped requirements to the solution functions. All pilot and solution requirements are included. This provides an overview of the current level of completeness of the requirements analysis and functional specification of the solution.
(3) The mapping results are analysed and discussed according to the evaluation criteria described in Section 2.3.7:
    - ***Completeness*** – The mapping diagram shows that 7 out of 9 sub-functions (lowest level) have mapped requirements, which corresponds to a completeness value of 0.78. The "blank spots" where no requirements are mapped in the diagrams could carry the risk of (still) unknown requirements not being fulfilled, while at the same time giving developers room to act because no specific requirements are given in the use case scenarios. There is also one “hot spot” in the diagram, the sub-functions “apply rules to identify harmful events” that has many mapped requirements to satisfy as it is one of the main functions of the solution.
    - ***Precision*** – The mapping diagram shows that 14 out of 16 requirements have been mapped to sub-functions (lowest level), which corresponds to a precision value of 0.88. It should be checked, if further specification by users (pilots) and solution providers is needed in the current requirements descriptions and their mapping.
    - ***Interface specification*** – In the FSD 2 input and zero output interfaces are identified and defined, namely i4Q_DR and i4Q_DIT. Since i4Q_IM should interact also with i4Q_AI and i4Q_EW, these interfaces should be added and clarified in the upcoming work packages 3 to 5 to ensure the interoperability.
    - ***Req Origin*** – The requirements have been defined by 8 different stakeholders. To ensure that this solution is created in such a way that many other i4Q Solutions can use it, collaboration with all affected stakeholders should be considered.

[24] This description is mainly taken from i4Q Deliverable 1.4 (Nowak-Meitinger and Jochem, 2021, p. 117).

The way i4Q_IM interacts with i4Q_AI and i4Q_EW has not been specified yet, since their interaction scheme is unknown at this stage of the project. The inclusion of other i4Q Solutions to the input of the i4Q_IM would be revisited in the development stage, where the requirements of the i4Q_AI and i4Q_EW solutions regarding pilots' needs will be further clarified.

The procedure for creating the diagrams and the purpose are described in Section 2. Further descriptions of the solution can be found in Deliverable 2.7 as well as in Deliverable 1.4. The functions of the solution are described in detail in Deliverable 2.5.

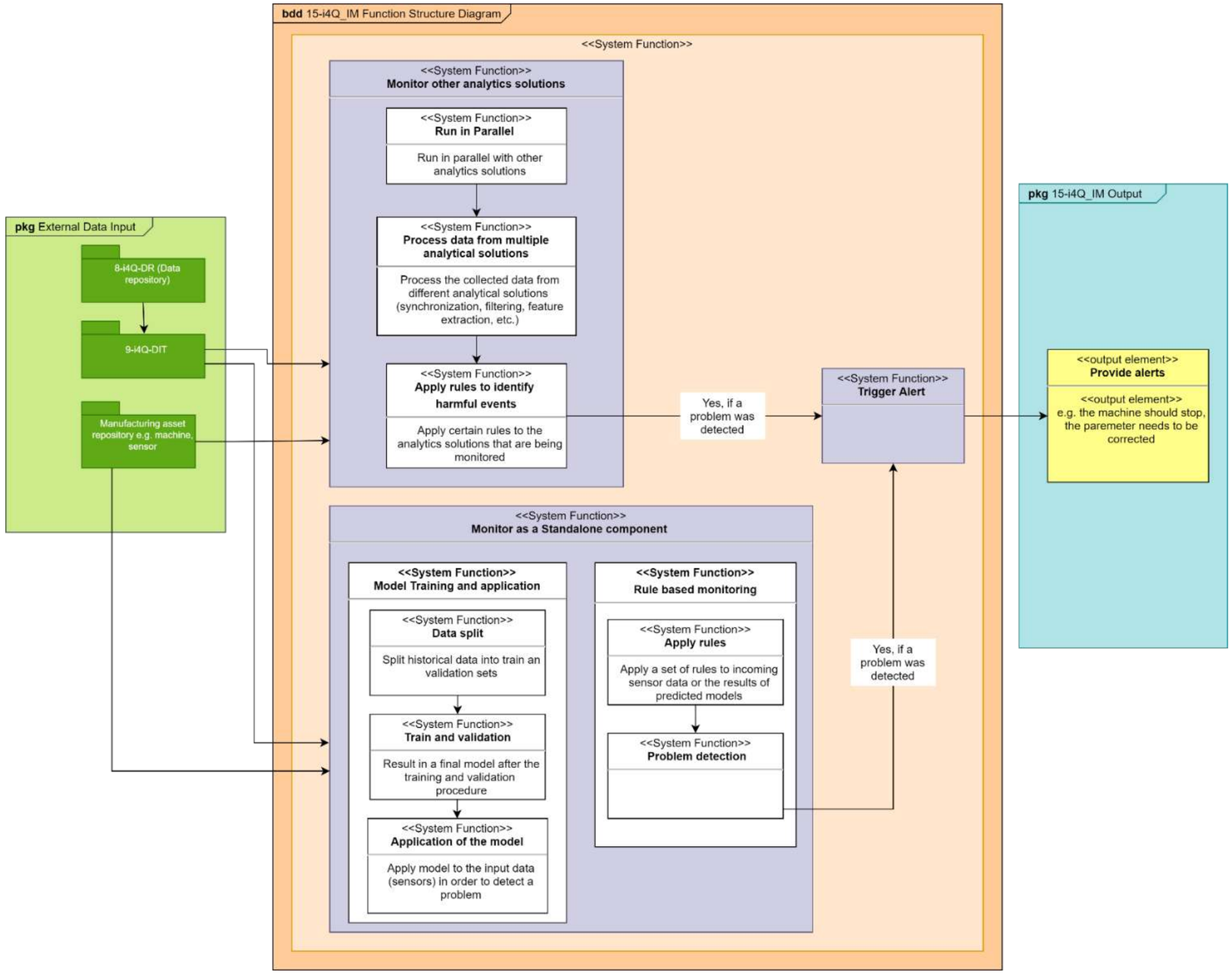


Figure 54. i4Q^{IM} Function Structure Diagram (FSD)

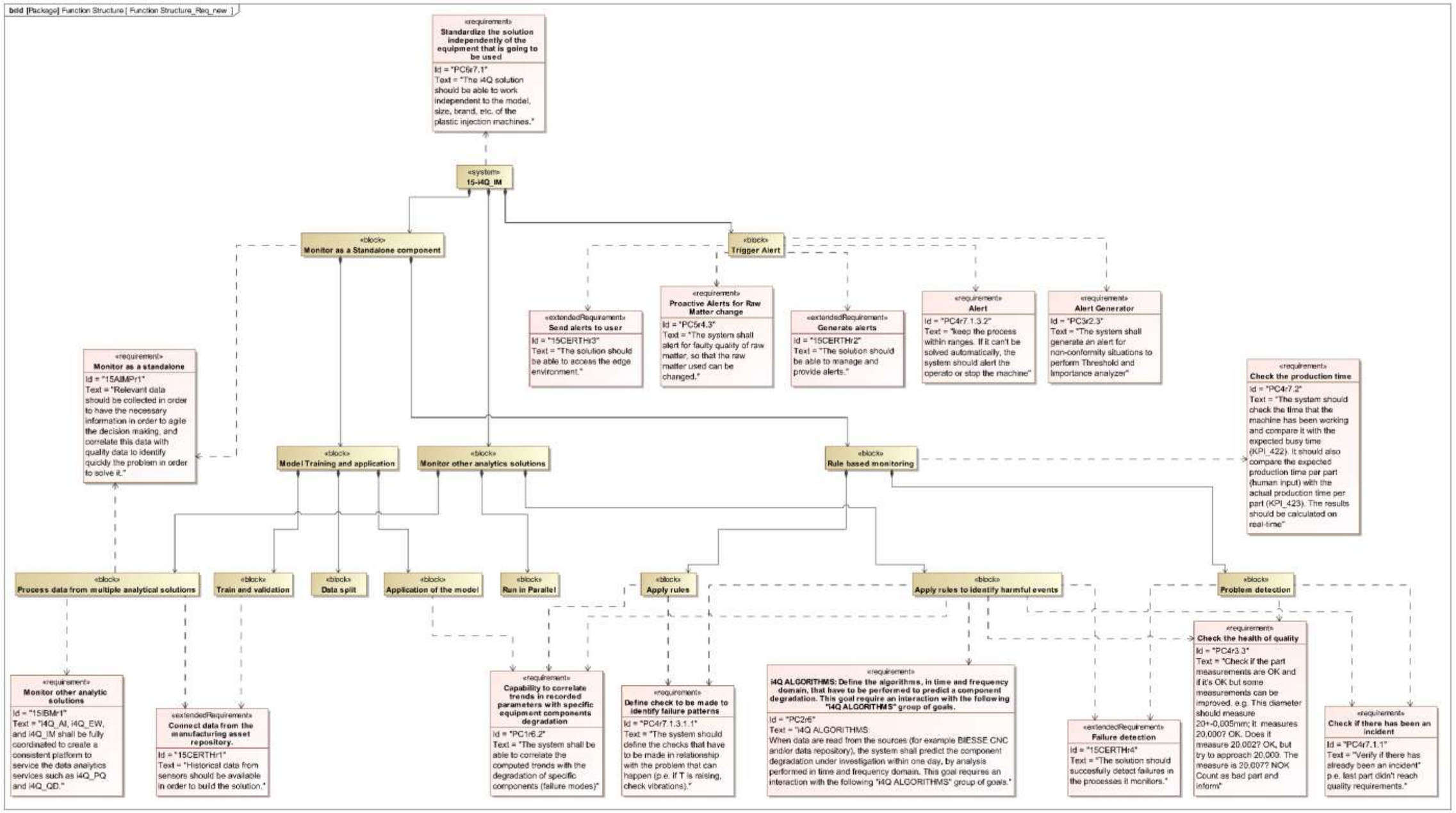


Figure 55. i4Q[IM] Requirements Mapping and Functional Specification

4.16 i4Q^{DT} Digital Twin Simulation Services

i4Q^{DT} is defined in Deliverable 1.4 as a system that provides a connected 3D production simulation environment. It enables virtual validation/visualisation and productivity optimisation. It will make use of both pre-existing data and simulated data (virtual sensors) obtained with the developed model. It will also make use of data from different factory levels (small cell to entire factory).[25]

The figures below provide the following information:

(1) Figure 56 shows the FSD of i4Q^{DT} with its inputs and outputs as well as the functions and sub-functions. Interfaces to other i4Q Solutions are also included. This provides a structural overview of the functionalities of the solution.
(2) Figure 57 shows the mapped requirements to the solution functions. All pilot and solution requirements are included. This provides an overview of the current level of completeness of the requirements analysis and functional specification of the solution.
(3) The mapping results are analysed and discussed according to the evaluation criteria described in Section 2.3.7:
    - *Completeness* – The mapping diagram shows that 14 out of 24 sub-functions (lowest level) have mapped requirements, which corresponds to a completeness value of 0.58. The "blank spots" where no requirements are mapped in the diagrams could carry the risk of (still) unknown requirements not being fulfilled, while at the same time giving developers room to act because no specific requirements are given in the use case scenarios.
    - *Precision* – The mapping diagram shows that 9 out of 11 requirements have been mapped to sub-functions (lowest level), which corresponds to a precision value of 0.82. It should be checked, if further specification by users (pilots) and solution providers is needed in the current requirements descriptions and their mapping.
    - *Interface specification* – In the FSD 3 input and 5 output interfaces are identified and defined. Since i4Q_DT should interact also with i4Q_AI and i4Q_EW, these interfaces should be added. All these interfaces should be then clarified in the upcoming work packages 3 to 5 to ensure the interoperability.
    - *Req Origin* – The requirements have been defined by 7 different stakeholders. To ensure that this solution is created in such a way that many other i4Q Solutions can use it, collaboration with all affected stakeholders should be considered.

The precision level observed in the solution is quite high. However, a couple of requirements have been mapped to the whole solution, as the available detail level of the solution. Therefore, when starting the development, a more detailed analysis would be needed to further specify them and make a clearer mapping. Moreover, the development phase of this solution will help

[25] This description is mainly taken from i4Q Deliverable 1.4 (Nowak-Meitinger and Jochem, 2021, p. 119).

to identify potential requirements for those sub-functions with no requirements mapped. Additionally, interactions with other solutions will need to be addressed in the next stages.

The procedure for creating the diagrams and the purpose are described in Section 2. Further descriptions of the solution can be found in Deliverable 2.7 as well as in Deliverable 1.4. The functions of the solution are described in detail in Deliverable 2.5.

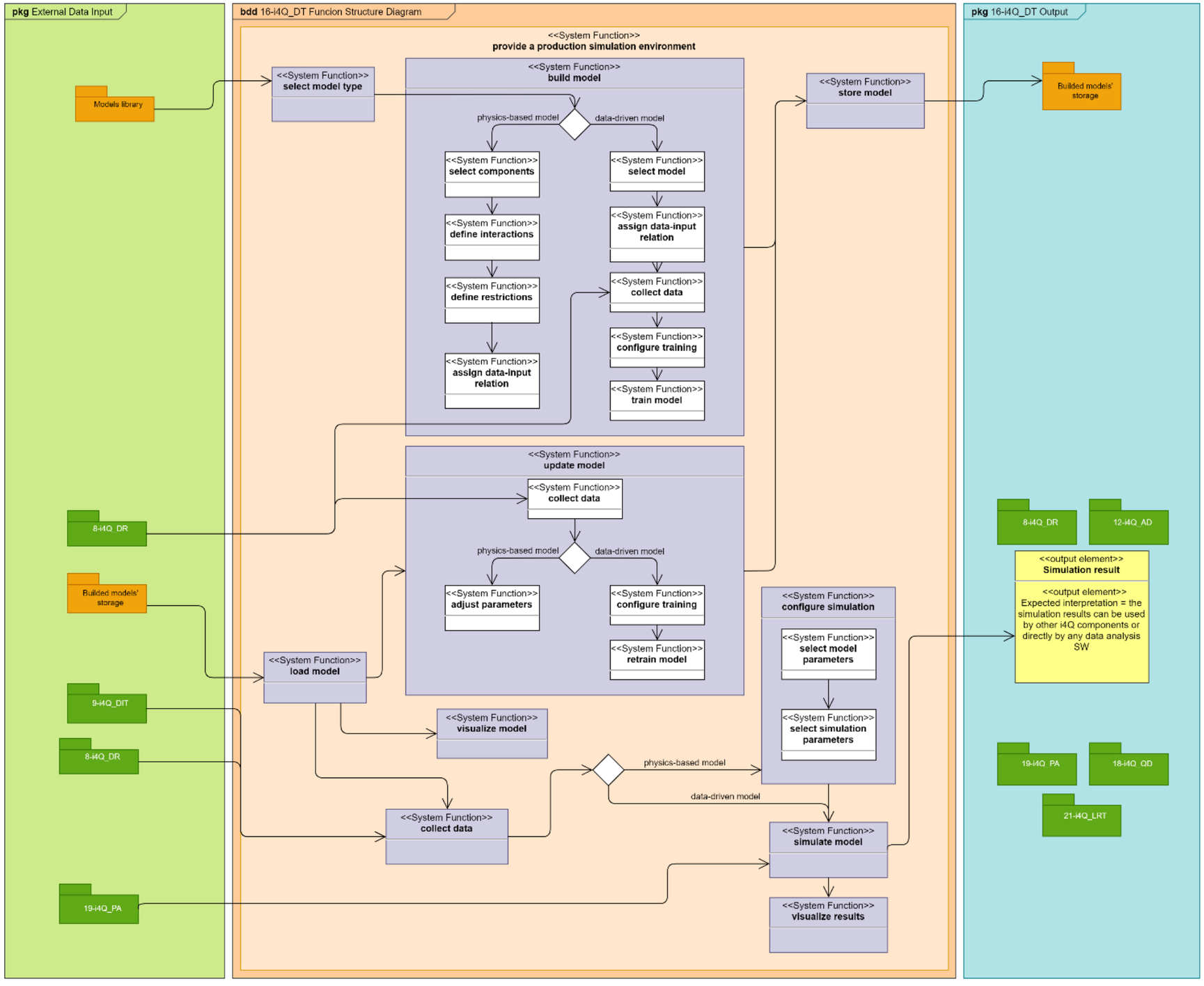


Figure 56. i4Q^{DT} Function Structure Diagram (FSD)

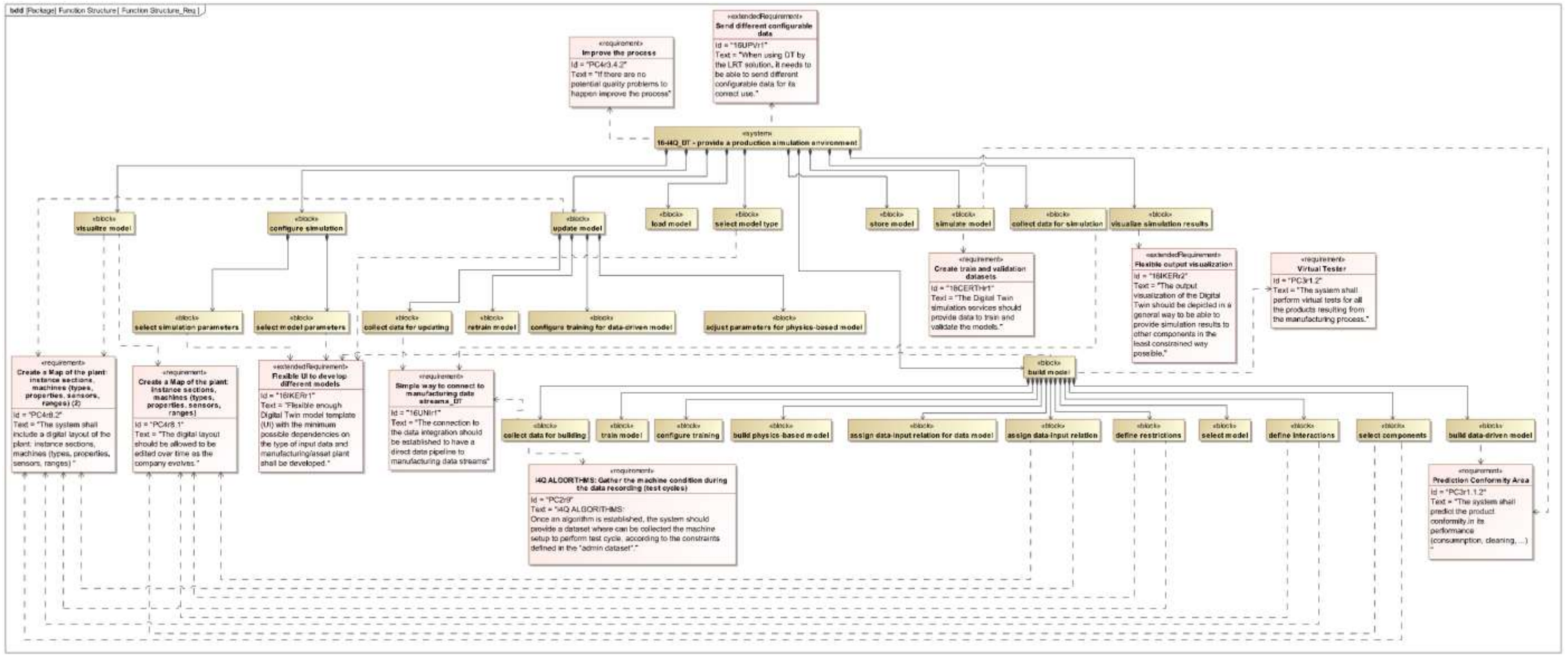


Figure 57. i4Q^{DT} Requirements Mapping and Functional Specification

4.17 i4Q^{PQ} Data-driven Continuous Process Qualification

i4Q^{PQ} is defined in Deliverable 1.4 as a system that provides a tool for achieving and ensuring process stability and product quality. Based on process data such as product quality, machine parameters, systematic influences, and interdependencies, an algorithm is developed. This algorithm can predict process stability and product quality. Therefore, it enables the process owner to get a deeper understanding how chances in the machine parameters will change the output of the process. As a result, e.g., the ramp-up time for a manufacturing process is reduced. By monitoring each step of the manufacturing process the product quality is predicted. If critical parameters change over time and pass a warning threshold, the process owner is informed and enabled to adjust the parameters to save setting before products with insufficient quality features are produced. The monitoring tool will be implemented as a dashboard solution.[26]

The figures below provide the following information:

(1) Figure 58 shows the FSD of i4Q^{PQ} with its input and outputs as well as the functions and sub-functions. Interfaces to other i4Q Solutions are also included. This provides a structural overview of the functionalities of the solution.
(2) Figure 59 shows the mapped requirements to the solution functions. All pilot and solution requirements are included. This provides an overview of the current level of completeness of the requirements analysis and functional specification of the solution.
(3) The mapping results are analysed and discussed according to the evaluation criteria described in Section 2.3.7:
    - *Completeness* – The mapping diagram shows that 2 out of 20 sub-functions (lowest level) have mapped requirements, which corresponds to a completeness value of 0.10. The "blank spots" where no requirements are mapped in the diagrams could carry the risk of (still) unknown requirements not being fulfilled, while at the same time giving developers room to act because no specific requirements are given in the use case scenarios. The “hot spots” in the diagram are the (sub-)functions “apply RNN to predict process quality data” and “evaluate process data”. These functions have many mapped requirements to satisfy. As “evaluate process data” is no sub-function on the lowest level its mapped requirements should be clarified precisely in the upcoming work packages.

[26] This description is mainly taken from i4Q Deliverable 1.4 (Nowak-Meitinger and Jochem, 2021, p. 122).

- *Precision* – The mapping diagram shows that 4 out of 12 requirements have been mapped to sub-functions (lowest level), which corresponds to a precision value of 0.33. It should be checked, if further specification by users (pilots) and solution providers is needed to be more precise in the current requirements descriptions and their mapping.
- *Interface specification* – In the FSD 4 input and 2 output interfaces are identified and defined. Since i4Q_PQ should also interact with i4Q_AI and i4Q_EW, these interfaces should be added. All interfaces should be then clarified in the upcoming work packages 3 to 5 to ensure the interoperability.
- *Req Origin* – The requirements have been defined by 5 different stakeholders. To ensure that this solution is created in such a way that many other i4Q Solutions can use it, collaboration with all affected stakeholders should be considered.

The unprecise mapping of some of the requirements has several reasons. Due to the unknown additional applications of the solutions, several specific requirements could not be mapped to low level functions. Furthermore, the description of the requirement has more information than the solution is currently able to deliver. Therefore, the requirements are mapped to the higher-level functions in the functional architecture which have to be addressed in the development process in WP5. This should improve both *Completeness* and *Precision*.

The procedure for creating the diagrams and the purpose are described in Section 2. Further descriptions of the solution can be found in Deliverable 2.7 as well as in Deliverable 1.4. The functions of the solution are described in detail in Deliverable 2.5.

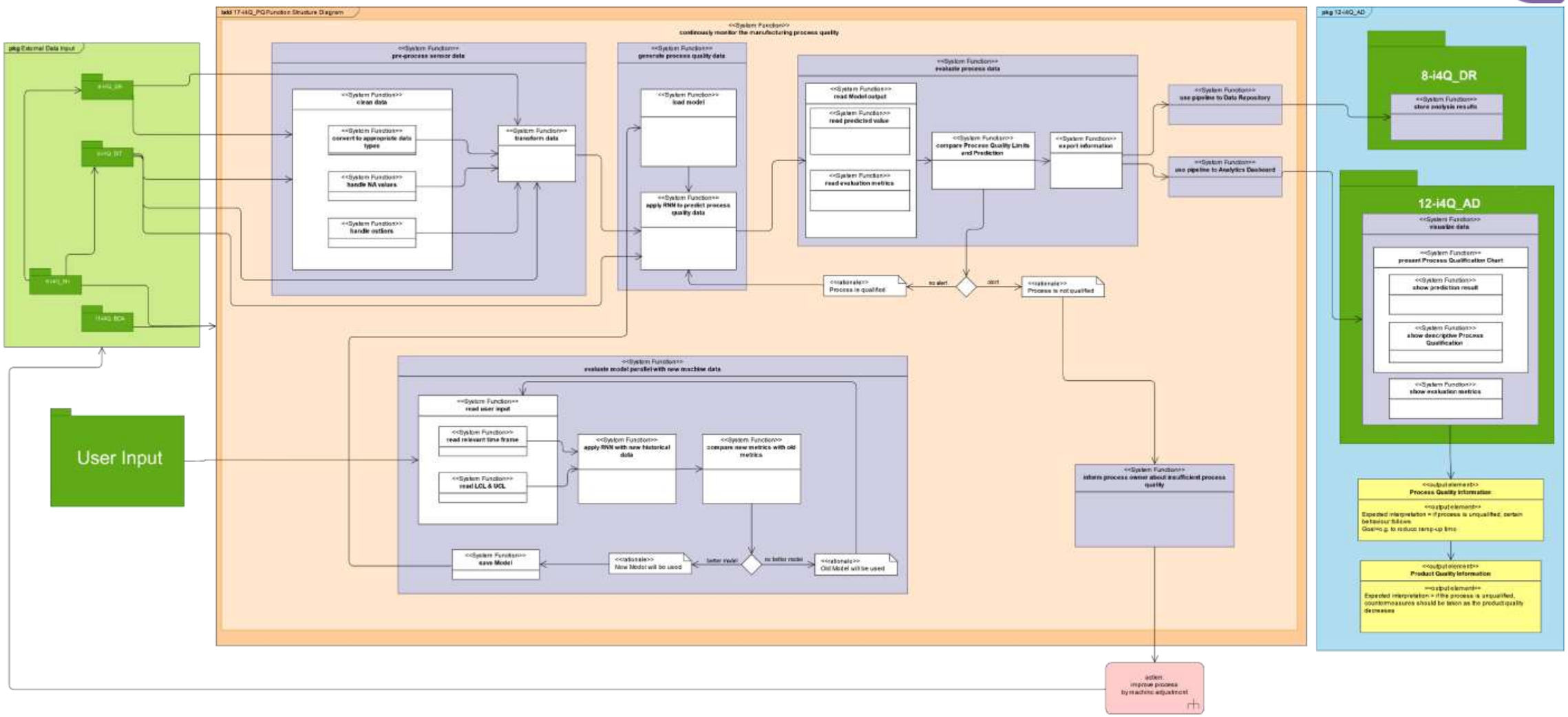


Figure 58. i4Q^{PQ} Function Structure Diagram (FSD)

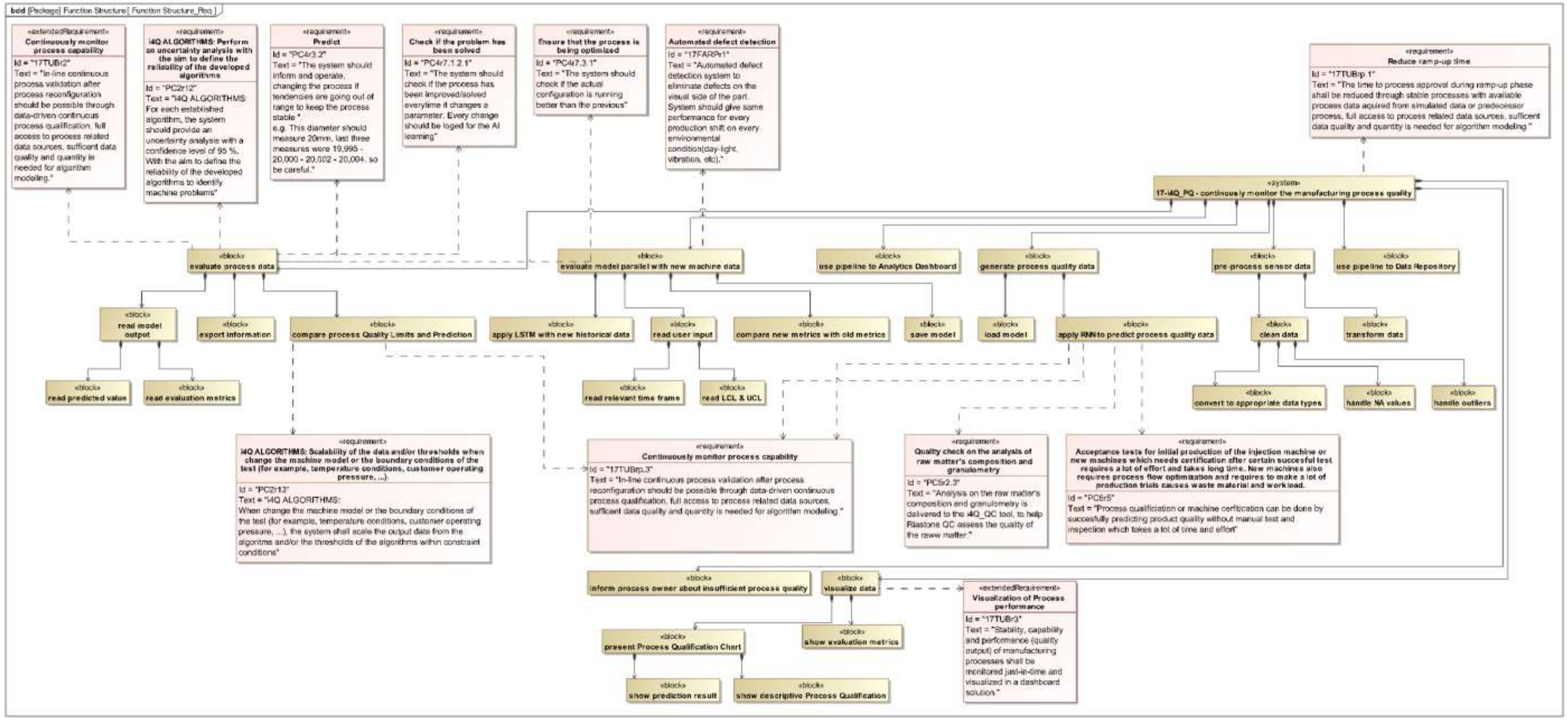


Figure 59. i4Q^{PQ} Requirements Mapping and Functional Specification

4.18 i4Q^{QD} Rapid Quality Diagnosis

i4Q^{QD} is defined in Deliverable 1.4 as a microservice whose main function is to monitor the manufacturing line and diagnose possible causes of failures. To ensure the correct functionality of the manufacturing line, the following aspects should be monitored by i4Q^{QD}: Evaluation of data, product quality, process condition and maintenance actions.[27]

The figures below provide the following information:

(1) Figure 60 shows the FSD of i4Q^{QD} with its input and outputs as well as the functions and sub-functions. Interfaces to other i4Q Solutions are also included. This provides a structural overview of the functionalities of the solution.
(2) Figure 61 shows the mapped requirements to the solution functions. All pilot and solution requirements are included. This provides an overview of the current level of completeness of the requirements analysis and functional specification of the solution.
(3) The mapping results are analysed and discussed according to the evaluation criteria described in Section 2.3.7:
    - ***Completeness*** – The mapping diagram shows that 4 out of 6 sub-functions (lowest level) have mapped requirements, which corresponds to a completeness value of 0.67. The "blank spots" where no requirements are mapped in the diagrams could carry the risk of (still) unknown requirements not being fulfilled, while at the same time giving developers room to act because no specific requirements are given in the use case scenarios. The sub-functions "Evaluation of the algorithm" and "Feature selection" have four or more mapped requirements to satisfy.
    - ***Precision*** – The mapping diagram shows that 13 out of 19 requirements have been mapped to sub-functions (lowest level), which corresponds to a precision value of 0.68. It should be checked, if further specification by users (pilots) and solution providers is needed in the current requirements descriptions and their mapping.
    - ***Interface specification*** – In the FSD 3 input and 1 output interfaces are identified and defined. These interfaces should be clarified in the upcoming work packages 3 to 5 to ensure the interoperability.
    - ***Req Origin*** – The requirements have been defined by 5 different stakeholders. To ensure that this solution is created in such a way that many other i4Q Solutions can use it, collaboration with all affected stakeholders should be considered.

Requirements that refer to general characteristics and functionalities of the i4Q^{QD} solution (e.g., "Interaction with other i4Q solutions") cannot be mapped to specific sub-functions, therefore resulting to low *Precision*. Additionally, sub-functions (e.g., "Spilt data") with too specialised functionalities depend on other sub-functions (e.g., "Train and validation") in order to fully

[27] This description is mainly taken from i4Q Deliverable 1.4 (Nowak-Meitinger and Jochem, 2021, p. 125).

satisfy a requirement (e.g., “Train different models”), thus affecting the *Completeness*. These mappings are unknown at this stage of the project. They are going to be resolved as the development of the i4Q_QD progresses and more detailed sub-functions are defined.

The procedure for creating the diagrams and the purpose are described in Section 2. Further descriptions of the solution can be found in Deliverable 2.7 as well as in Deliverable 1.4. The functions of the solution are described in detail in Deliverable 2.5.

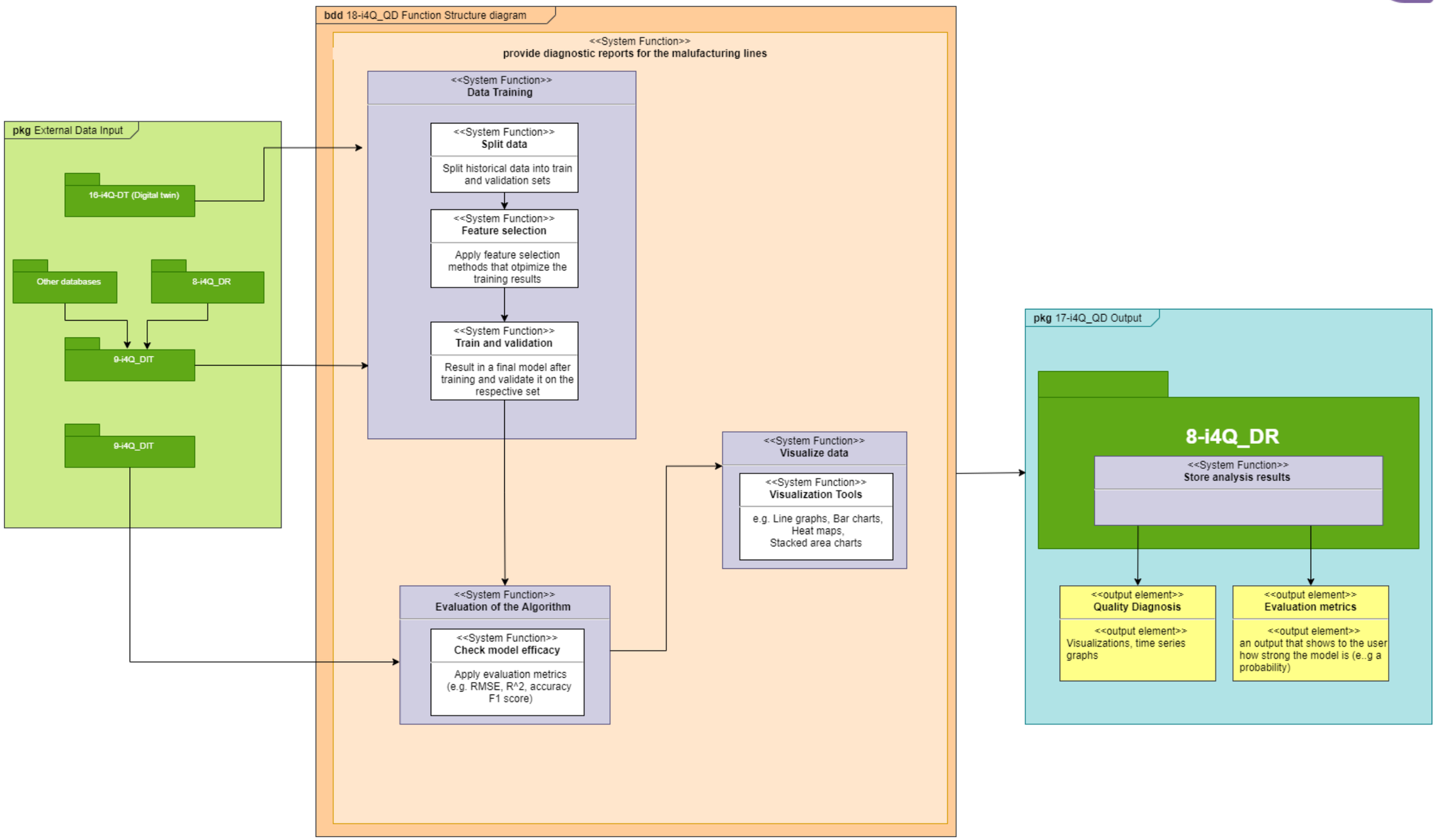


Figure 60. i4Q^{QD} Function Structure Diagram (FSD)

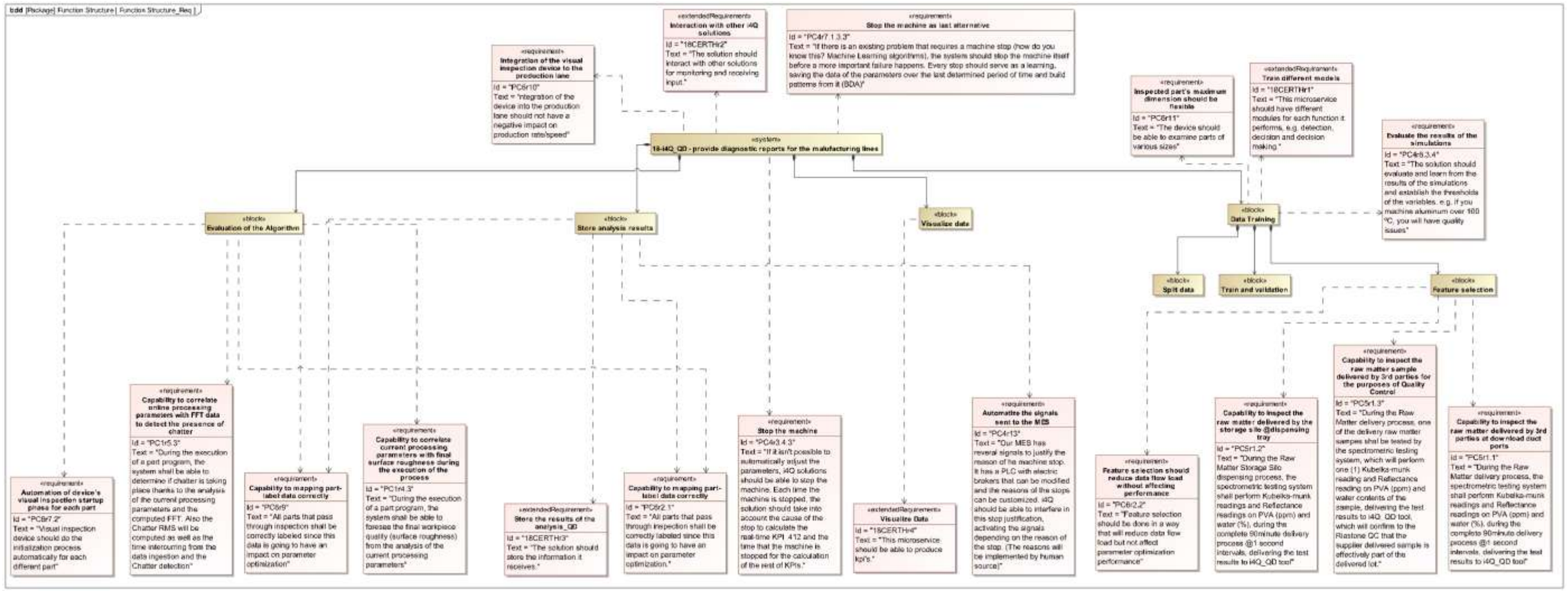


Figure 61. i4Q^{QD} Requirements Mapping and Functional Specification

4.19 i4Q^{PA} Prescriptive Analysis Tools

i4Q^{PA} is defined in Deliverable 1.4 as a system that provides mainly simulations as a service. It exploits the Digital Twin developed to test different configuration parameters and analyze the effect small changes can produce in the production. It makes use of manufacturing resources, production planning and process condition. The prescriptive analysis will come from exhaustive simulation with no specific optimization algorithm implemented.[28]

The figures below provide the following information:

(1) Figure 62 shows the FSD of i4Q^{PA} with its input and outputs as well as the functions and sub-functions. Interfaces to other i4Q Solutions are also included. This provides a structural overview of the functionalities of the solution.
(2) Figure 63 shows the mapped requirements to the solution functions. All pilot and solution requirements are included. This provides an overview of the current level of completeness of the requirements analysis and functional specification of the solution.
(3) The mapping results are analysed and discussed according to the evaluation criteria described in Section 2.3.7:
    - ***Completeness*** – The mapping diagram shows that 6 out of 11 sub-functions (lowest level) have mapped requirements, which corresponds to a completeness value of 0.55. The "blank spots" where no requirements are mapped in the diagrams could carry the risk of (still) unknown requirements not being fulfilled, while at the same time giving developers room to act because no specific requirements are given in the use case scenarios.
    - ***Precision*** – The mapping diagram shows that 6 out of 8 requirements have been mapped to sub-functions (lowest level), which corresponds to a precision value of 0.75. It should be checked, if further specification by users (pilots) and solution providers is needed in the current requirements descriptions and their mapping.
    - ***Interface specification*** – In the FSD 3 input and 2 output interfaces are identified and defined. These interfaces should be clarified in the upcoming work packages 3 to 5 to ensure the interoperability.
    - ***Req Origin*** – The requirements have been defined by 3 different stakeholders. To ensure that this solution is created in such a way that many other i4Q Solutions can use it, collaboration with all affected stakeholders should be considered.

Regarding the *Precision*, there are no requirements mapped to the whole solution, which indicates that the requirements have such a detail level that it has been possible to associate them at least to some higher-level functions. However, in some cases the lower-level functions may be too specific, which generates some difficulties when defining requirements for them. Anyway, the blank spots identified are not considered critical, as most of them are necessary

[28] This description is mainly taken from i4Q Deliverable 1.4 (Nowak-Meitinger and Jochem, 2021, p. 131).

steps for other sub-functions. Additionally, interactions with other solutions will need to be accounted for in the next steps.

The procedure for creating the diagrams and the purpose are described in Section 2. Further descriptions of the solution can be found in Deliverable 2.7 as well as in Deliverable 1.4. The functions of the solution are described in detail in Deliverable 2.5.

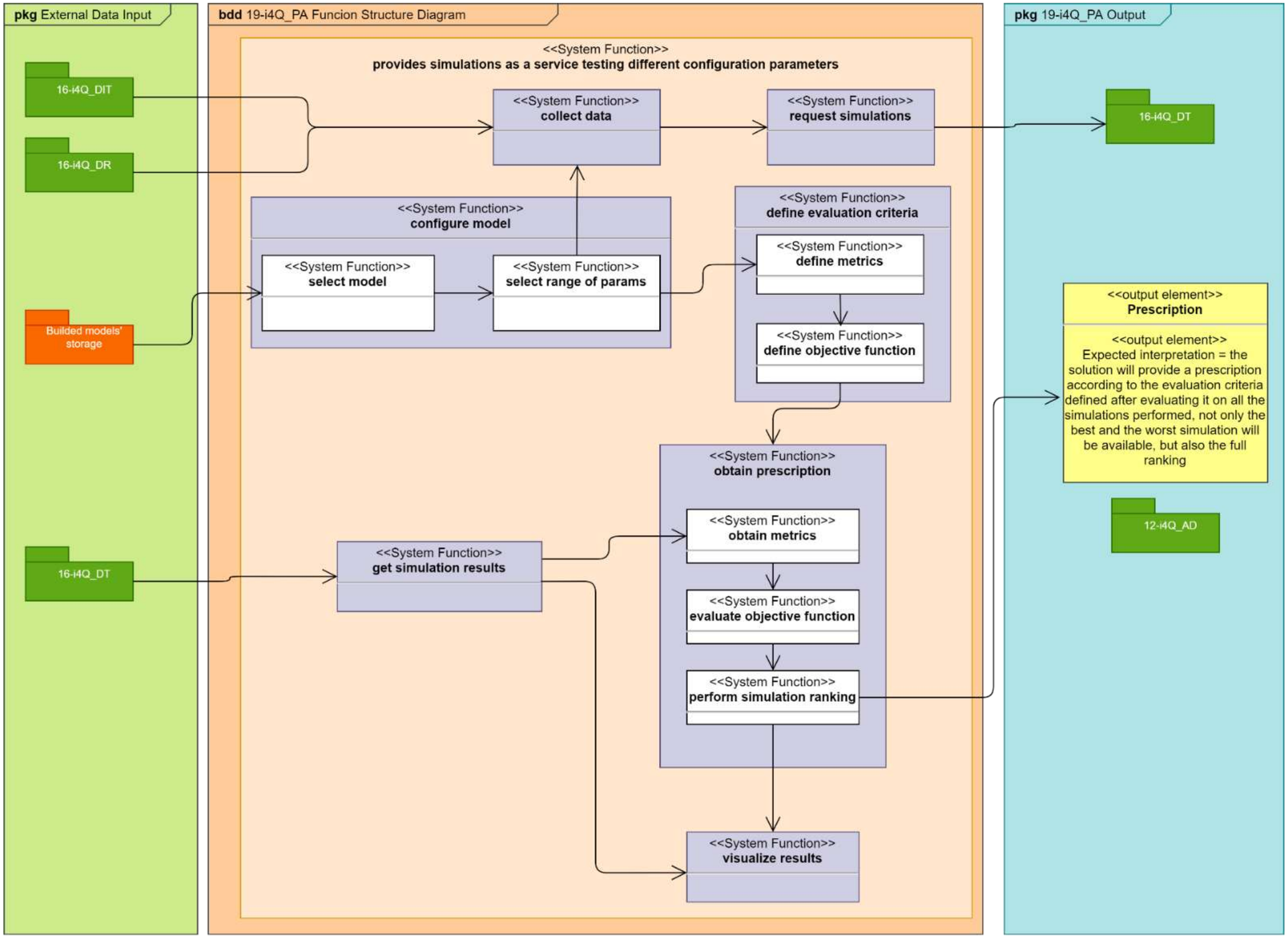


Figure 62. i4Q[PA] Function Structure Diagram (FSD)

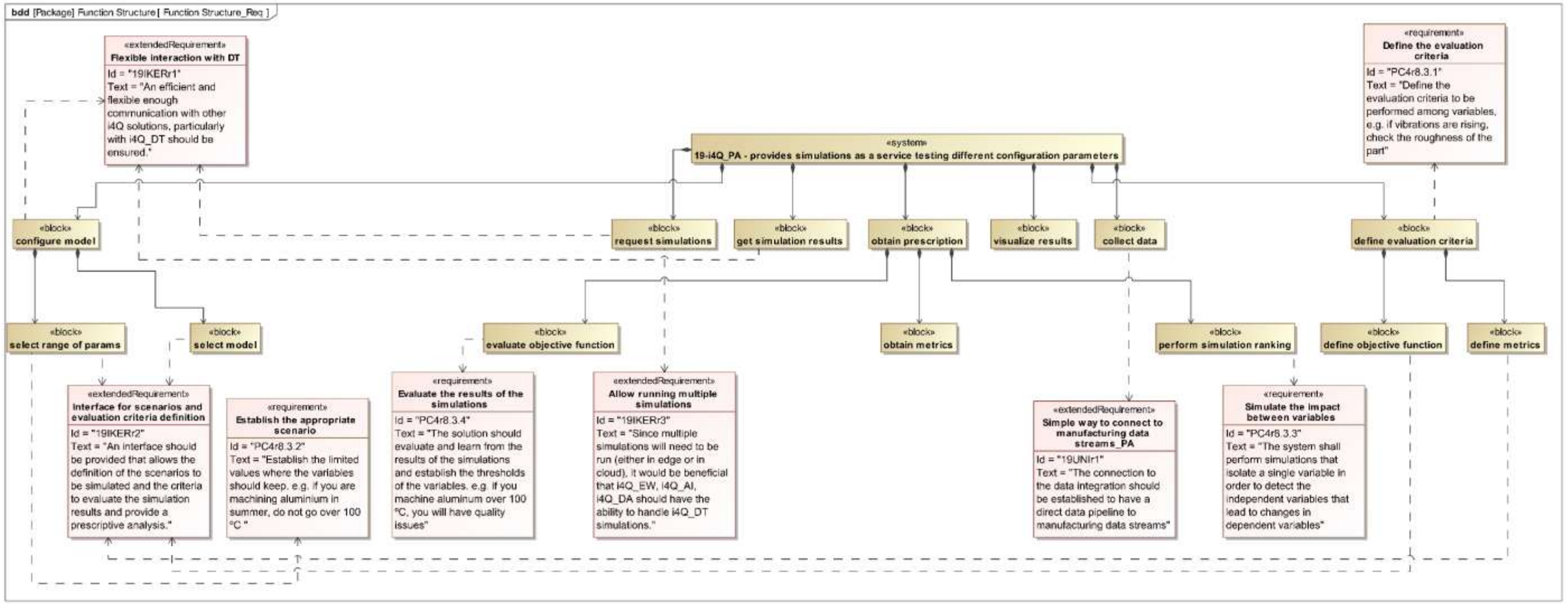


Figure 63. i4Q^{PA} Requirements Mapping and Functional Specification

4.20 i4Q[LRG] Manufacturing Line Reconfiguration Guidelines

i4Q[LRG] is defined in Deliverable 1.4 as a tool that provides a multi-media guide to reconfiguration the manufacturing line. To do so, it considers the results obtained when calibrating the parameters of the machine. Based on this information, this guideline will ensure long-term productivity. In addition, it will reduce the reconfiguration efforts of the manufacturing line.[29]

After a revision of all requirements in this second deliverable of the requirements analysis and functional specification task, the following requirements are set up for the i4Q[LRG] (Table 5).

ID	Type of Requirement	Requirement Title	Requirement definition (related to the i4Q Solution)	Requirement coming from connected Solution
20UPVr1	Guidelines req, Usability and Quality req	User guide about Manufacturing Line reconfiguration	An interactive and intuitive guide for users about Manufacturing Line reconfiguration should be provided. (No specific requirements from additional i4Q Solutions.)	20-i4Q_LRG
20UPVr2	Guidelines req, Usability and Quality req	Technical document	A technical document of the manufacturing line (it will be intuitive) should be provided.	20-i4Q_LRG
20UPVr3	Functional/Technical req, Usability and Quality req	Store and facilitate historical query data	This solution should store historical query data to facilitate its use to the end-user.	20-i4Q_LRG

Table 5. Requirements mapped to i4Q[LRG]

4.21 i4Q[LRT] Manufacturing Line Reconfiguration Toolkit

i4Q[LRT] is defined in Deliverable 1.4 as a toolkit whose objective is to increase productivity and reduce the efforts for manufacturing line reconfiguration through AI. This tool consists of a set of services that will allow to analyse and simulate different scenarios and provide the best configuration for the modules and parameters of the manufacturing line.[30]

The figures below provide the following information:

[29] This description is mainly taken from i4Q Deliverable 1.4 (Nowak-Meitinger and Jochem, 2021, p. 134).
[30] This description is mainly taken from i4Q Deliverable 1.4, (Nowak-Meitinger and Jochem, 2021, p. 135).

(1) Figure 64 shows the FSD of i4Q[LRT] with its input and outputs as well as the functions and sub-functions. Interfaces to other i4Q Solutions are also included. This provides a structural overview of the functionalities of the solution.

(2) Figure 65 shows the mapped requirements to the solution functions. All pilot and solution requirements are included. This provides an overview of the current level of completeness of the requirements analysis and functional specification of the solution.

(3) The mapping results are analysed and discussed according to the evaluation criteria described in Section 2.3.7:

- *Completeness* – The mapping diagram shows that 3 out of 15 sub-functions (lowest level) have mapped requirements, which corresponds to a completeness value of 0.20. The "blank spots" where no requirements are mapped in the diagrams could carry the risk of (still) unknown requirements not being fulfilled, while at the same time giving developers room to act because no specific requirements are given in the use case scenarios.
- *Precision* – The mapping diagram shows that 3 out of 14 requirements have been mapped to sub-functions (lowest level), which corresponds to a precision value of 0.21. It should be checked, if further specification by users (pilots) and solution providers is needed to be more precise in the current requirements descriptions and their mapping.
- *Interface specification* – In the FSD 4 input and zero output interfaces are identified and defined. These interfaces should be clarified in the upcoming work packages 3 to 5 to ensure the interoperability.
- *Req Origin* – The requirements have been defined by 5 different stakeholders. To ensure that this solution is created in such a way that many other i4Q Solutions can use it, collaboration with all affected stakeholders should be considered.

The precision level observed in the solution is quite high. It is a solution with different applications, from the construction of distributable files for the different algorithms, to the use of these. This creates some difficulties in defining the requirements for them, which can be interpreted as very generic. There are a number of functions and subfunctions that are not explicitly mapped to requirements, but they are assumed to be needed, in order to properly support other mapped functions in the solution. Thanks to distributable files, the algorithms can be distributed around the edge within the industry. Allowing different systems or tools to use them locally, reducing latency. In addition, it will have access to different local resources (either data or sensors). The inclusion of other i4Q Solutions to the input and output of the i4Q_LRT would be revisited in the development stage, in particular, interaction with digital twin-type tools and solutions.

The procedure for creating the diagrams and the purpose are described in Section 2. Further descriptions of the solution can be found in Deliverable 2.7 as well as in Deliverable 1.4. The functions of the solution are described in detail in Deliverable 2.5.

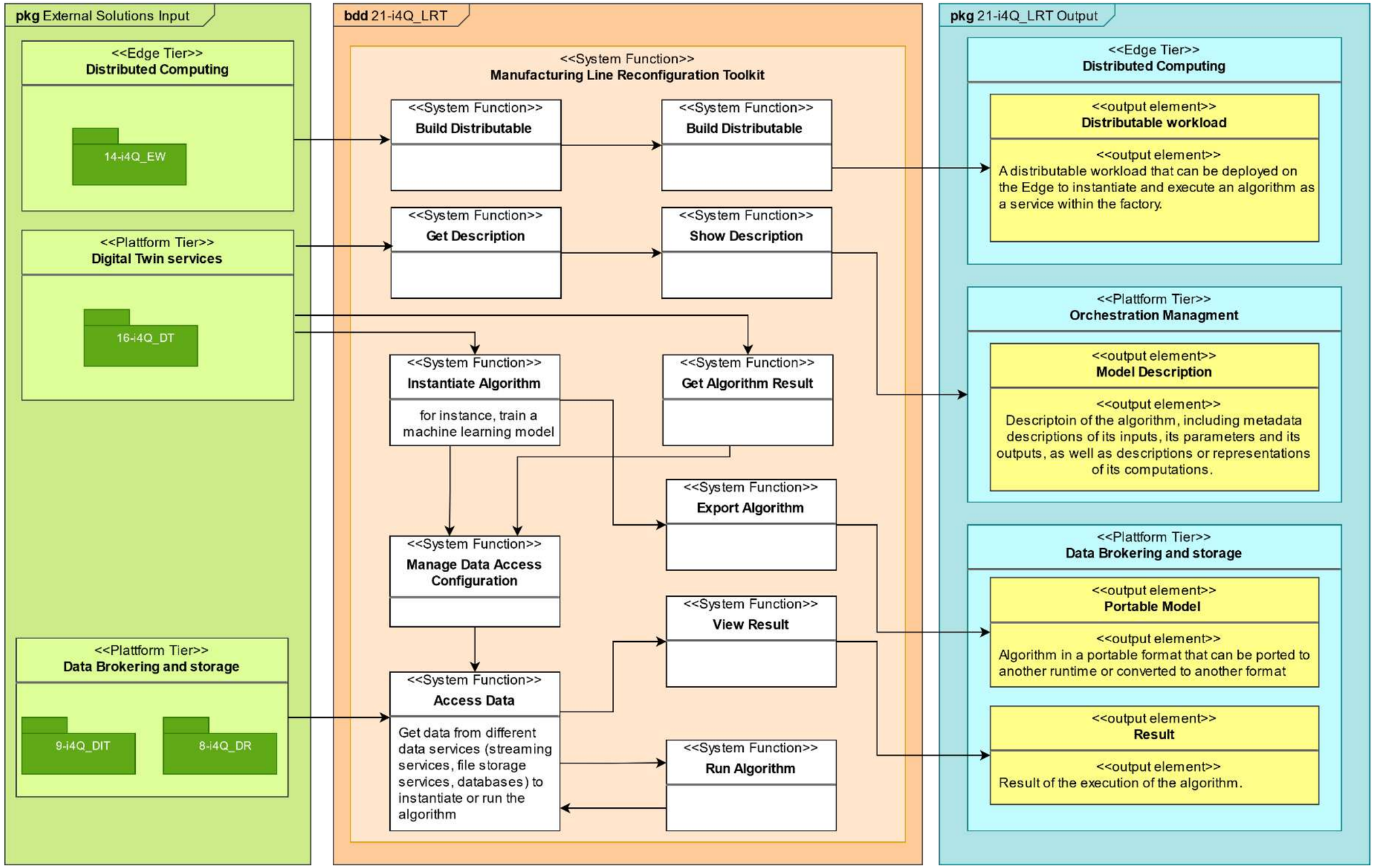


Figure 64. i4Q[LRT] Function Structure Diagram (FSD)

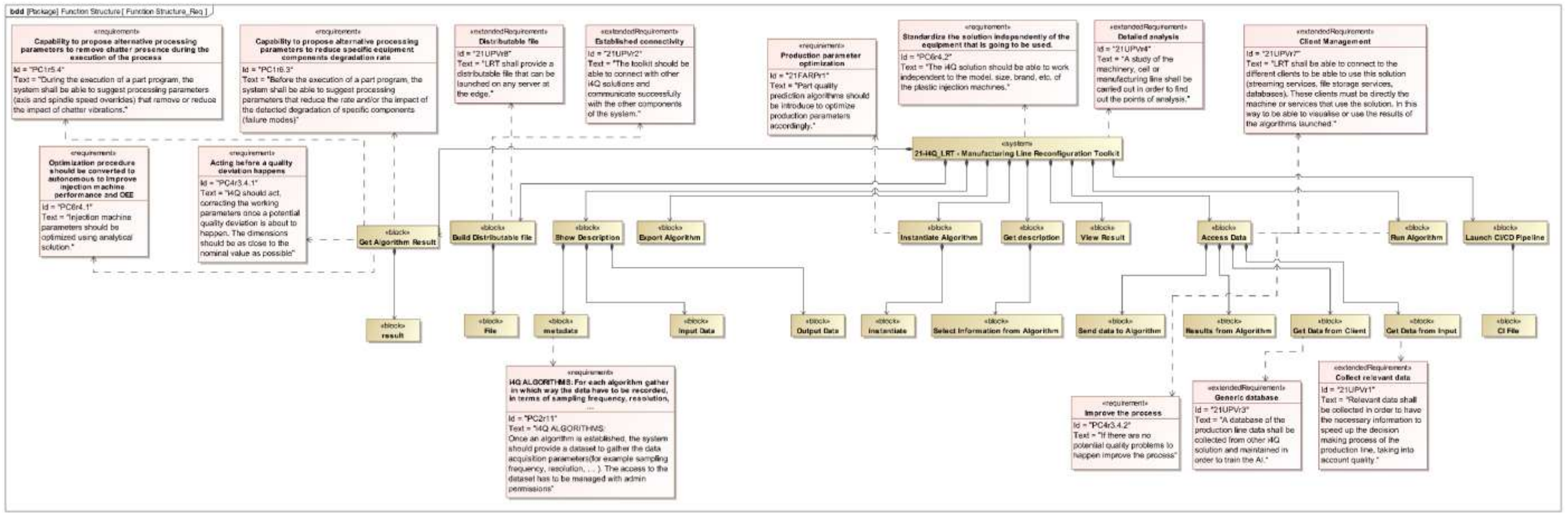


Figure 65. i4Q[LRT] Requirements Mapping and Functional Specification

4.22 i4Q^{LCP} Manufacturing Line Data Certification Procedure

i4Q^{LCP} is defined in Deliverable 1.4 as a guideline with a related digital assistant. The guideline describing the certification procedure is based on standards and further guidelines. i4Q^{LCP} will support standardization and implementation of data processes in manufacturing lines considering the Service Oriented Architecture (SoA) and the state with implemented i4Q Solution. Based on this information, a workflow is developed and transferred into a procedure. The procedure contains all necessary steps a company needs to perform in order to ensure the full functionality of all i4Q Solutions. The guideline and procedure comprise are the standard audit documents for ensuring long term functionality and ensure stable and performant processes and product quality. The digital assistant is a so-called conversational AI. It supports users via natural language communication during the audit. The assistant will provide, for instance, information about audit steps and it will allow users to monitor audit progress. The assistant integrates the functionality of the QualiExplore i4Q^{QE} solution and may also interact with other i4Q Solutions relevant to an audit. The digital assistant incorporates the information described in the guideline. This concerns workflows and checklists. Since building the assistant relies on training data, it will require example conversations.[31]

The figures below provide the following information:

(1) Figure 66 shows the FSD of i4Q^{LCP} with its input and outputs as well as the functions and sub-functions. Interfaces to other i4Q Solutions are also included. This provides a structural overview of the functionalities of the solution.
(2) Figure 67 shows the mapped requirements to the solution functions. All pilot and solution requirements are included. This provides an overview of the current level of completeness of the requirements analysis and functional specification of the solution.
(3) The mapping results are analysed and discussed according to the evaluation criteria described in Section 2.3.7:
    - *Completeness* – The mapping diagram shows that 0 out of 9 sub-functions (lowest level) have mapped requirements, which corresponds to a completeness value of 0.0. This lack of mapping ("blank spots") could carry the risk of (still) unknown requirements not being fulfilled, while at the same time giving developers room to act because no specific requirements are given in the use case scenarios. Since this solution is mainly a guideline no functional/technical requirements are mapped. Nevertheless, all necessary requirements should be clarified in the upcoming work package 5.
    - *Precision* – The mapping diagram shows that 0 out of 3 requirements have been mapped to sub-functions (lowest level), which corresponds to a precision value of 0.0. It should be checked, if further specification by users (pilots) and solution

[31] This description is mainly taken from i4Q Deliverable 1.4, (Nowak-Meitinger and Jochem, 2021, p. 139).

providers is possible in order to specify the current requirements descriptions and their mapping more precisely.

- *Interface specification* – In the FSD 1 interface to i4Q_QE is identified and defined. This interface should be then clarified in the upcoming work packages 3 and 5 to ensure the interoperability.
- *Req Origin* – The requirements have been defined by 2 different stakeholders. To ensure that this solution is created in such a way that many other i4Q Solutions can use it, collaboration with all affected stakeholders should be considered.

Since this solution is built to be a guideline for an audit procedure after the implementation of the i4Q Solutions, all functions of this solutions are made to ensure the overarching certification process. As advisory, i4Q_LCP is strongly connected to i4Q_QE. Therefore, additional coordination has to be established in WP5.

The procedure for creating the diagrams and the purpose are described in Section 2. Further descriptions of the solution can be found in Deliverable 2.7 as well as in Deliverable 1.4. The functions of the solution are described in detail in Deliverable 2.5.

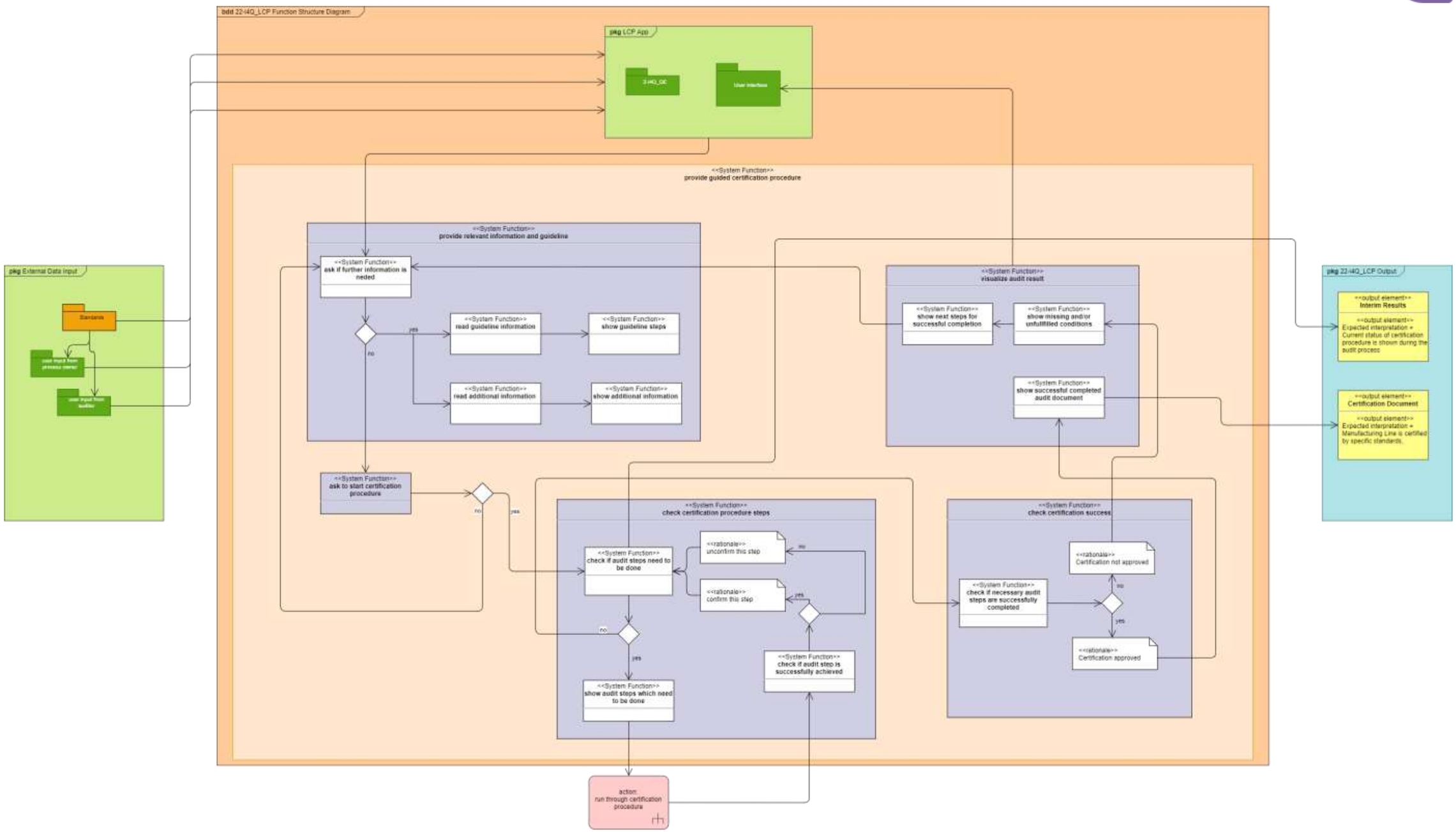


Figure 66. i4Q^{LCP} Function Structure Diagram (FSD)

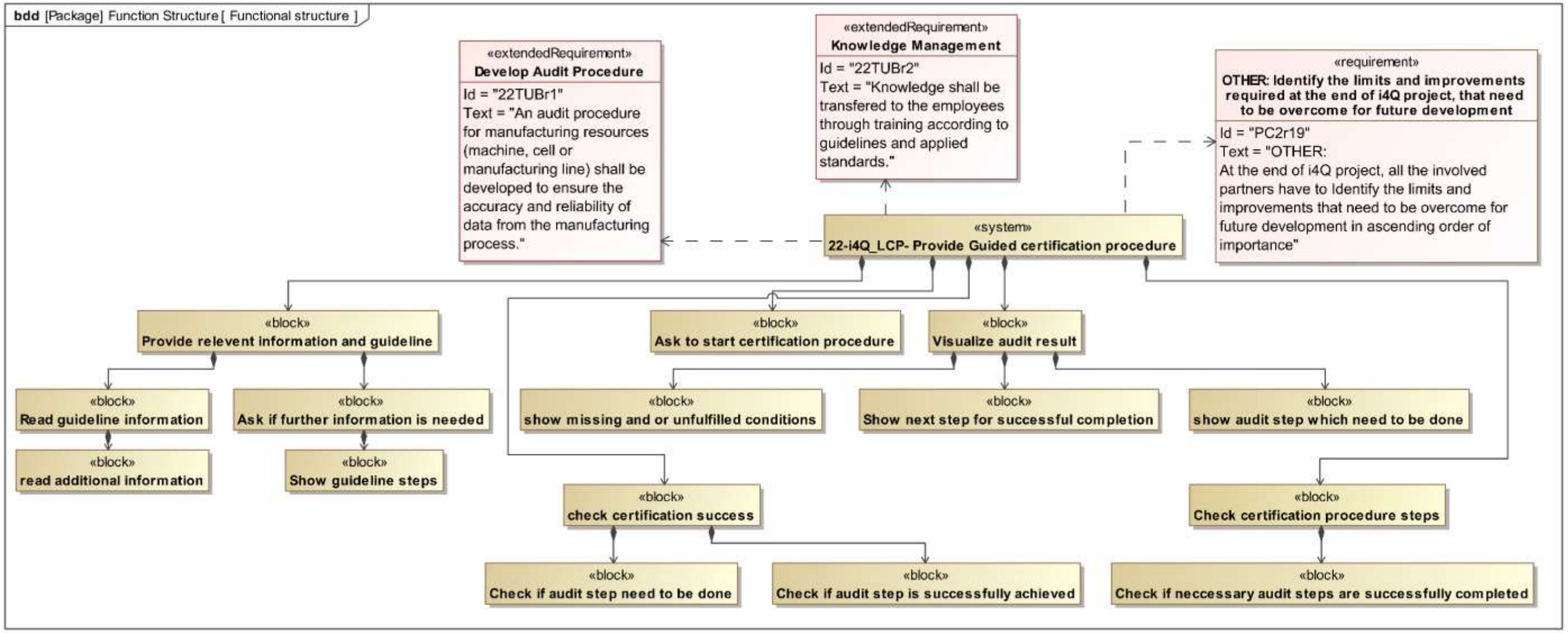


Figure 67. i4Q[LCP] Requirements Mapping and Functional Specification

4.23 Unmapped requirements

Table 6 shows the requirements that could not be mapped to any i4Q solution. The simple reason for this inconsistency is that the functionality of the i4Q Solutions cannot satisfy these requirements properly. In the special case of PC2r15 and PC2r20 where a user interface is required, the i4Q partner discussions came to an agreement to not map these requirements to any i4Q Solution. This gap has to be addressed in the BUILD work packages where the functionality is developed in close relation to BIESSE. Another set of requirements which was left unmapped is part of FARP who requires hardware robustness which consists of the adaptability to different environmental conditions like dust, light and temperature. Since all technical i4Q Solutions only provide software, these requirements remain unmapped. Nevertheless, these requirements can be redirected to all pilot partners and are crucial in the implementation WP 6.

ID	Type of Requirement	Requirement Title	Requirement definition (related to the i4Q Solution)
PC2r15	Usability and Quality req	Biesse support (hotline or filed) should be able to send feedback to the i4Q platform, with the aim of increasing the reliability of the i4Q solutions	The system should provide a "user interface" to collect feedback from hotline or field, with the aim to increase the reliability of the i4Q algorithms.
PC2r20	Usability and Quality req	Building an interface where the users can define the parameters to perform the required analysis	Once an algorithm is established, the system should provide an interface where the test cycles can be designed according to admin and user dataset.
PC6r6	Functional/Technical req	Adaptability to different environmental conditions should be provided	Since the environmental conditions (e.g. lightness, temperature, etc.) can vary from day-to-day the process should not be affected by environmental factors.
PC6r6.1	Functional/Technical req	Adaptability to different environmental conditions should be provided	Quality control unit should not be affected by dust.
PC6r6.2	Functional/Technical req	Adaptability to different environmental conditions should be provided	Quality control unit should not be affected by light.
PC6r6.3	Functional/Technical req	Adaptability to different environmental conditions should be provided	Quality control unit should not be affected by environmental heat up to 40° Celsius.

Table 6. Unmapped requirements

4.24 Requirements to the complete set of i4Q RIDS

Table 7 consists of the general requirements mapped to the entire set of i4Q RIDS to complete the set of proposed requirements. Three of the listed requirements below were stated by TUB and are related to digital process transformation, traceability of defect, and data acquisition process storage and access respectively. Two IBM requirements are related to adaptability of edge computing platform and blockchain infrastructure, and the general AIMP requirements are dealing with the adaptability of the i4Q Solutions. Technical requirements from ITI are based on the improvement of data exchange and its associated technologies, and the requirement from TIAG addresses the broad integration of IEE 802.1 TSN.

The requirements listed in the table are generic and vague, so as consequence, the direct mapping to any of the i4Q Solutions is not useful. They should be rather seen as overarching goals the solutions should provide and how the output of the project can be qualitatively measured.

ID	Type of Require-ment	Requirement Title	Requirement definition (related to the i4Q Solution)
GEN_TUBr1	General Req	Digital process transformation	Data analytics and data-driven approaches should support through complete digital transformation of process related data.
GEN_TUBr2	General Req	Traceability of defect	Automatic Traceability of defects should be clear, consistent and continuous labelling of all components and products throughout the value-added process.
GEN_TUBr3	General Req	Data Acquisition, process, storage and access	Industrial partners should use data acquisition, processing, storage and accessing technologies.
GEN_IBMr1	General Req	Adaptability of edge computing platform	The development of edge computing platform should be addressing a broad set of current and future requirements from industrial and manufacturing use cases. The edge platform is expected to address a wide range of aspects as defined by i4Q. IBM's contributions will address aspects of enabling intelligent data processing at the manufacturing floor by bringing AI/ML to the edge close to where the data is being generated. Having these capabilities at the edge will significantly improve the quality, reliability, and efficiency in manufacturing.
GEN_IBMr2	General Req	Trust and traceability Blockchain infrastructure	The development of blockchain based infrastructure should be addressing data trust and traceability in industrial and manufacturing use cases. The blockchain platform is expected to address cornerstone issues of data reliability to enable smart processing by analysis components, as well as establishing trust among partners.

ID	Type of Require-ment	Requirement Title	Requirement definition (related to the i4Q Solution)
			Having these capabilities will significantly enhance the quality, reliability, and trust in manufacturing data.
GEN_AIMPr 1	General Req	Adaptability of i4Q Solutions	The i4Q Solution should be adapted to a variety of different machine types (injection machine makers, year of manufacturing, format of the data process to collect...).
GEN_ITIr1	General Req	Improve data exchange	Improve the reliability and quality of data exchange in the plant by deploying a new generation of private infrastructure based on 5G networks. These networks should offer a highly scalable and independent architecture that increases traffic capacity and network efficiency compared to 4th generation technologies.
GEN_ITIr2	General Req	Improvement in communication latency	Improvement in communication latency should be achieved through various techniques and standardised mechanisms, such as TSN, to achieve wireless communication have a latency similar to wired communications.
GEN_ITIr3	General Req	Introduce WPAN environment	Integrate the software-defined networking paradigm into a WPAN environment with TSCH MAC. This allows to implement dynamic and fully centralized resource provisioning, increasing flexibility and scalability as well as allowing node mobility while reducing management complexity.
GEN_TIAGr1	General Req	Broad integration of IEE 802.1 TSN	IEE 802.1 TSN (deterministic wired communication) should be integrating into a wide collection of machines and increase the use of deterministic communication into industry.

Table 7. Requirements to the complete set of i4Q RIDS

5. Evaluation and Use of Results

5.1 Evaluation of the Iteration Process

The procedure described in Section 2.3 has been implemented successfully. The methodology, which is based on established methods and standards (see Section 2.2) and tailored to the i4Q Project, can be evaluated as appropriate, since the four iteration steps could be implemented according to plan and led to useful results.

The main challenge was to keep track of and coordinate the multitude of use cases and solutions. Also, defining the interfaces between the i4Q Solutions to ensure the interoperability of the i4Q RIDS is a constant challenge. Furthermore, the exchange to other tasks and deliverables to be developed in parallel has led to an effort to avoid duplications and to use synergies. The goal of this deliverable was to create prerequisites for the DESIGN work package, especially for the Functional and Implementation Viewpoint.

During the four iteration steps, the requirements of the pilots and the functional specifications of the i4Q Solutions have been elicited and collected. The exchange between solution providers and end-users (pilots) during the four iteration steps has led to a precise mapping of the requirements to the specific functions of the solutions. During this process, the pilots' requirements diagrams and Function Structure Diagrams (FSDs) have served as communication tools and have led to a productive exchange of information between the partners. After the iterative creation of the SysML diagrams, the solution providers were thus able to create an initial functional specification for their solutions, including the FSD and the mapped requirements.

5.2 Evaluation of Functional Specification Results

To attain insights about the i4Q Solutions in relation to each other considering all written requirements, the results of the previous sections, the complete set of i4Q RIDS is evaluated. Figure 68 shows the *Requirement Origin Comparison* across all i4Q Solutions. Pilot requirements are highlighted in purple; requirements from Solution Providers – including their own – are in grey. With 38 requirements, i4Q^{DR} is the most addressed among all i4Q Solutions. Furthermore, i4Q^{DR} has a high number of interactions with other i4Q Solutions. Thus, this solution seems to have high importance within the project. Similar importance can be observed for i4Q^{DIT} where most mappings were made among pilot requirements. Other highly mentioned i4Q Solutions are dealing with services of data analytics like i4Q^{DA}, i4Q^{BDA}, i4Q^{AD} or i4Q^{QD}. The least mentioned requirements were stated for i4Q^{BC}, i4Q^{AI}, and i4Q^{EW}. Since i4Q^{LCP} is a guideline but using technical parts of other i4Q Solutions, it is included but not treated as part of the overall evaluation.

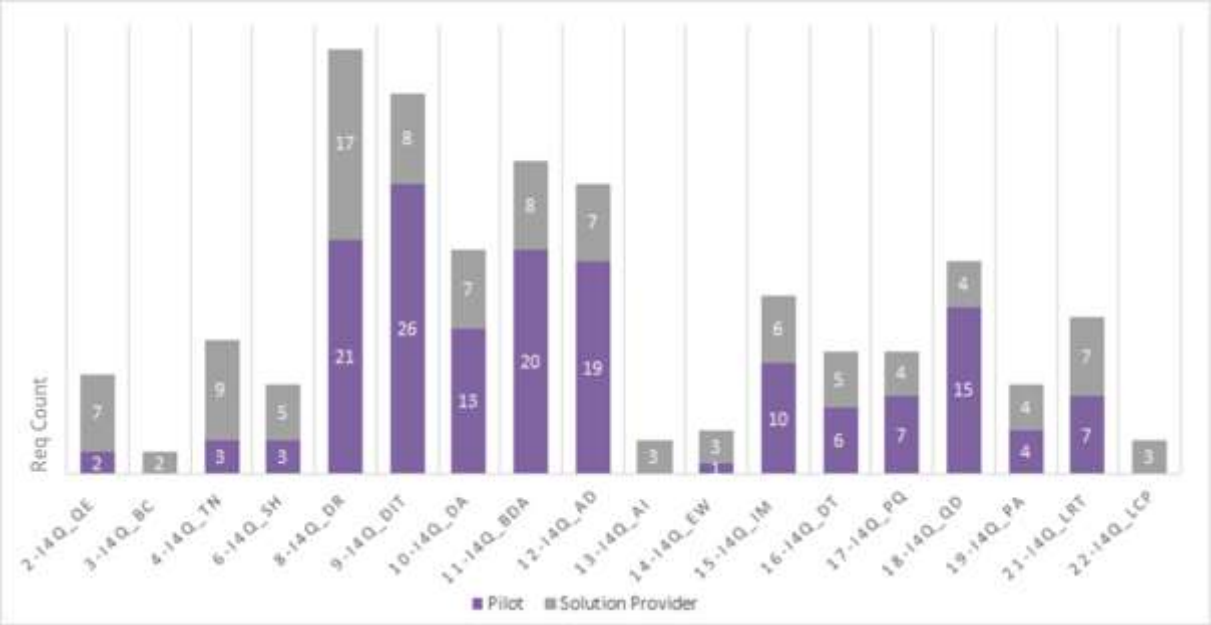
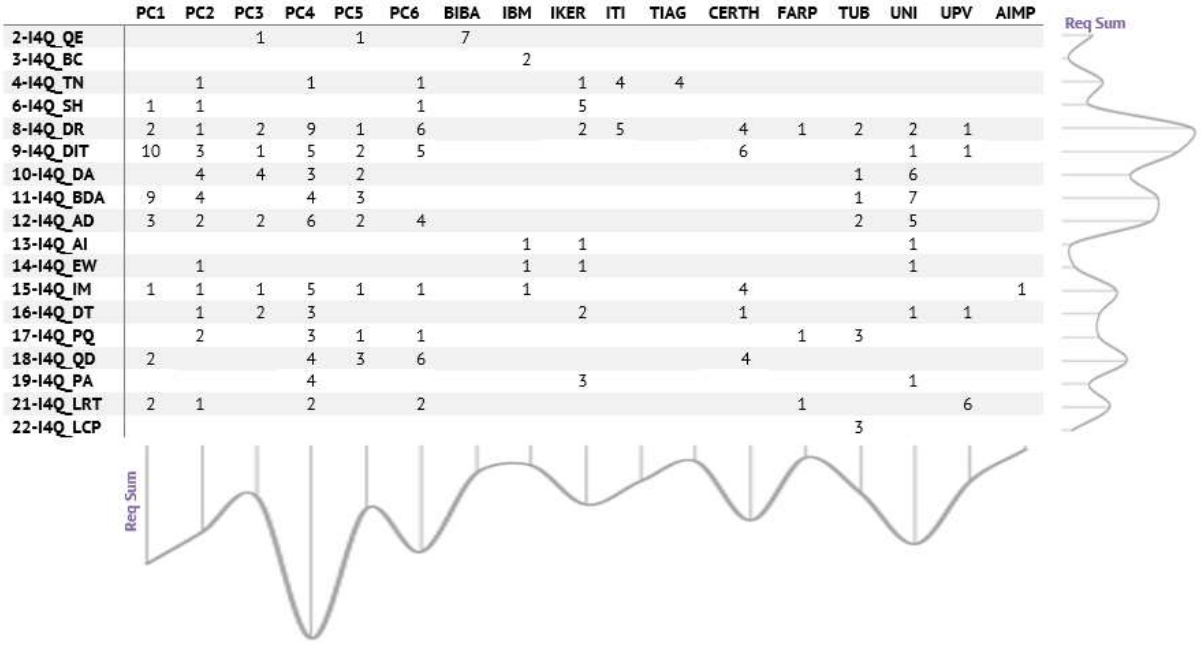


Figure 68. Requirements Origin Comparison

As a more detailed insight, Figure 69 shows the distribution of pilot and solution requirements across all i4Q Solutions. Therefore, the interaction between stakeholders and i4Q Solutions is quantified. This enables deeper knowledge of interactions for Solution Providers and Pilots. Among all pilots, FACTOR as PC4 has the highest number of requirements stated to i4Q Solutions. Furthermore, both BIESSE as PC2 and FACTOR show most interactions with i4Q Solutions. The least interactions needed can be observed for WHIRLPOOL as PC3. Interesting to see is the high interest of FIDIA as PC1 in $i4Q^{DIT}$ and $i4Q^{BDA}$. Analogously, FACTOR addressed most of their requirements to $i4Q^{DR}$.

Similar as above, the number of stakeholder requirements includes also requirements which were assigned by Solution Providers to their own solution. Nevertheless, UNI and IKER seem to have the highest number interaction with their technologies among all Solution Providers. Therefore, the definition of interfaces is crucial during the BUILD WPs 3 to 5 and further in the implementation.



	PC1	PC2	PC3	PC4	PC5	PC6	BIBA	IBM	IKER	ITI	TIAG	CERTH	FARP	TUB	UNI	UPV	AIMP
2-I4Q_QE			1		1		7										
3-I4Q_BC								2									
4-I4Q_TN		1		1		1			1	4	4						
6-I4Q_SH	1	1				1			5								
8-I4Q_DR	2	1	2	9	1	6			2	5		4	1	2	2	1	
9-I4Q_DIT	10	3	1	5	2	5						6			1	1	
10-I4Q_DA		4	4	3	2									1	6		
11-I4Q_BDA	9	4		4	3									1	7		
12-I4Q_AD	3	2	2	6	2	4								2	5		
13-I4Q_AI								1	1						1		
14-I4Q_EW		1						1	1						1		
15-I4Q_IM	1	1	1	5	1	1		1				4					1
16-I4Q_DT		1	2	3					2			1			1	1	
17-I4Q_PQ		2		3	1	1							1	3			
18-I4Q_QD	2			4	3	6						4					
19-I4Q_PA				4					3						1		
21-I4Q_LRT	2	1		2		2							1			6	
22-I4Q_LCP														3			

Figure 69. Distribution Table of Pilot and Solution Requirements across all i4Q Solutions

As overall analysis result, the classification procedure helps to differentiate between single i4Q Solutions regarding their challenges and opportunities concerning all mapped requirements. Its detailed content description about the fields in the chart and the used evaluation criteria can be found in section 2.3.7. The application of this approach is shown by the visualization of the *Status Comparison* in Figure 70. The colour of the bubbles has no specific expression and is chosen randomly, the bubble size correlates with the number of mapped requirements to the specific solution. Multiple i4Q Solutions have been classified as convenient for the development process, being part of the left-upper field respectively. The two i4Q Solutions in the right-upper field are i4Q^{DT} and i4Q^{DR}. Since it has a higher number of interactions a comprehensive software development process regarding the related stakeholders is recommended for this field. Having less interfaces but also a less specified requirements mapping, the i4Q Solutions classified in the left-lower field need more attention concerning the detailed description of requirements in BUILD WPs 3 to 5. The i4Q Solution which has most interactions but is less specified concerning the requirements for its low-level system functions is i4Q^{SH}. Due to the high number of interfaces, this i4Q Solution can contain a higher risk for the positive outcome of the project. Additionally, if the solution is not well defined by the stakeholders, the development process can be time consuming and even produce a mismatch concerning the usage of the technology. As special notice concerning the number of interfaces, the technology which is used for i4Q^{DR} is actually clear in its usability. Nevertheless, most of the requirements are mapped to a small number of low-level system functions. Additionally, it has the highest number of mapped requirements, given the large bubble size. This has to be addressed in the BUILD WP 3 with additional organizational effort to fulfil the stakeholders' needs.

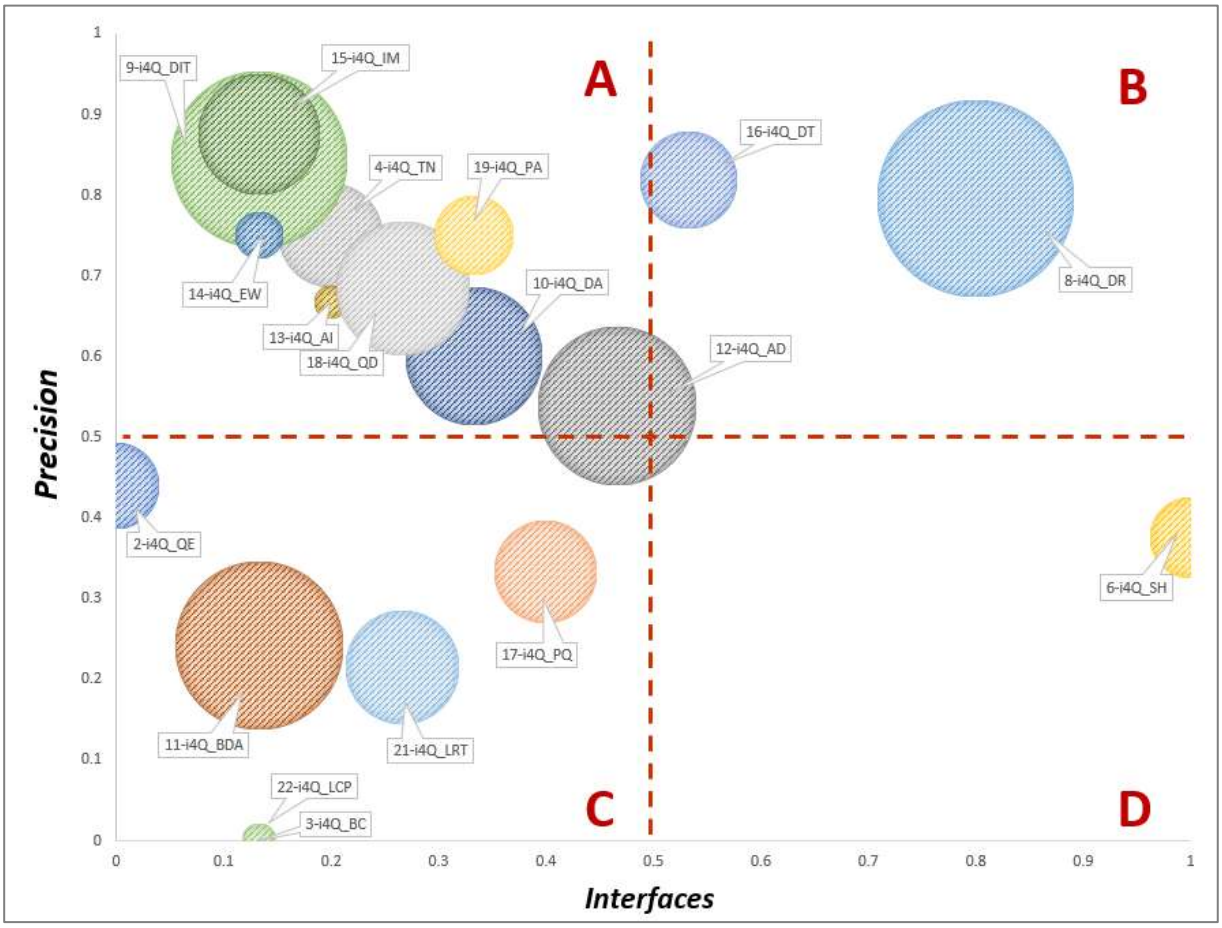


Figure 70. Status Comparison of all i4Q Solutions

For all software solutions, FSDs have been created containing the various interfaces between the solutions. The solution map in Figure 71 shows all input and output connections (interfaces) of the i4Q Solutions found in the FSD and explanations of Section 4. Solutions that are not connected to others are shown in the blocks below the diagram. The evaluation of all FSD and the *i4Q Solution Map* show that the interfaces between i4Q Solutions should be further analysed and updated in the upcoming work packages of the BUILD phase. There are some inconsistencies in the diagrams that should be discussed and deleted. In some cases, the i4Q Solutions (e.g., BC, AI, EW) should be more integrated in the map of all i4Q Solutions, i.e., interfaces should be added and specified.

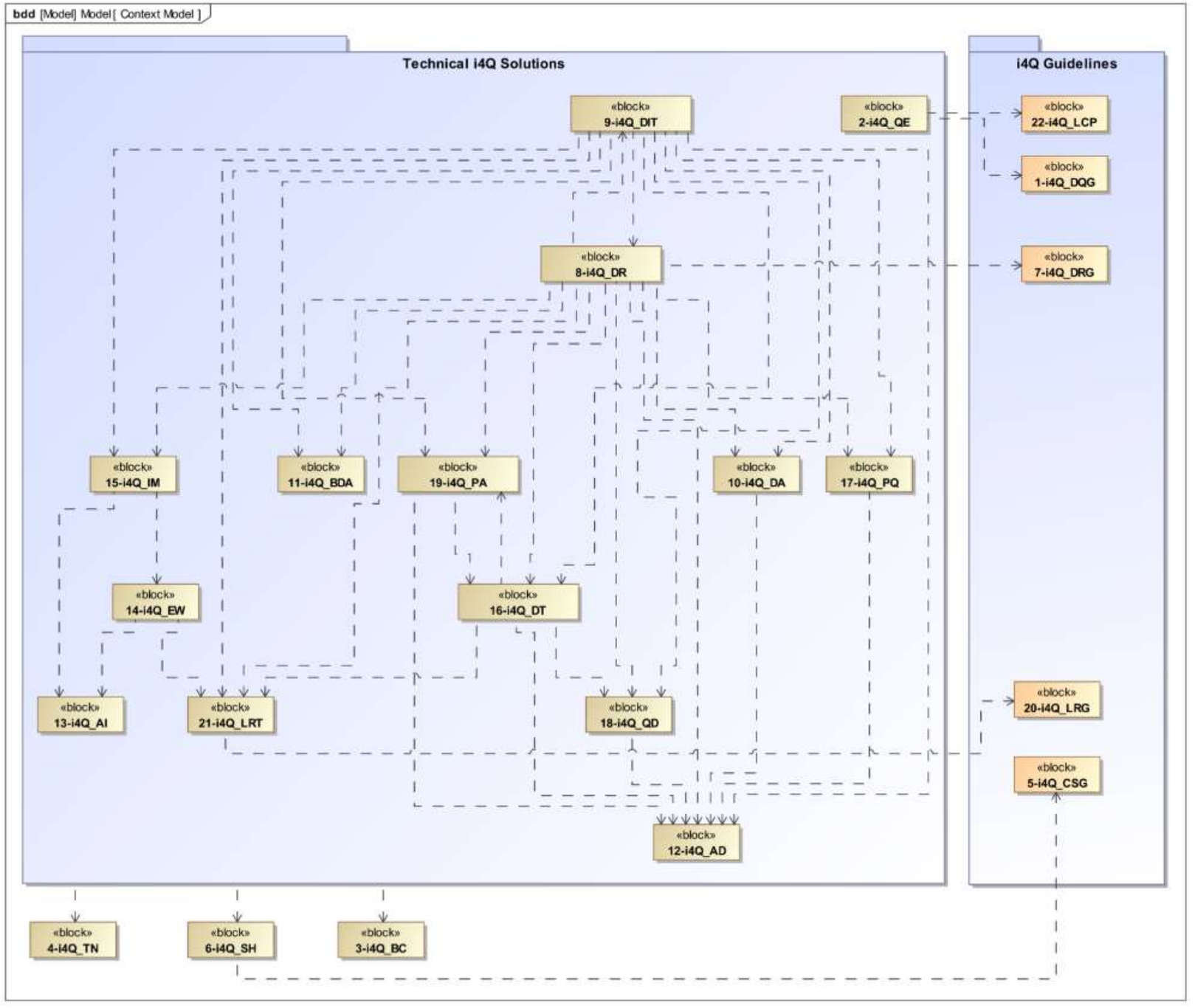


Figure 71. i4Q Solution Map

5.3 Use of Results in work packages 3 to 6

The results of this deliverable serve as a basis for the work in the upcoming work packages 3 to 6. The Functional Specification of the i4Q Solutions provided in this deliverable consists of the FSD and the mapped requirements as well as an evaluation, explanation and outlook for the further work. This provides the solution providers with a direction for the development of their solutions in the upcoming BUILD work packages. As shown in Figure 72, the new V-model of VDI 2206 (VDI2206:2020-09) demands requirements management, i.e., the handling of requirements, such as the structuring, mapping, analysis, and integration of requirements changes throughout all the work packages (Graessler and Hentze 2020, p. 321). In this sense, the outcomes of D1.9 will help solution providers to build the solutions in WP3, WP4 and WP5 by verifying if all the mapped requirements to the system sub-functions are fulfilled for the respective solution and, if necessary, changing, refining, or complementing the requirements.

In work package 6, as the experimental base for the i4Q Solutions, functional specifications with mapped pilot requirements and solution system requirements can be used for further verification and validations of the results. Once the solutions are built in WP3, WP4, WP5, the individual system elements and sub-systems have to be implemented, evaluated and validated in the pilot factories. Validation can be carried out against the mapped requirements. The system can be said to be validated if each requirement is verified and validated.

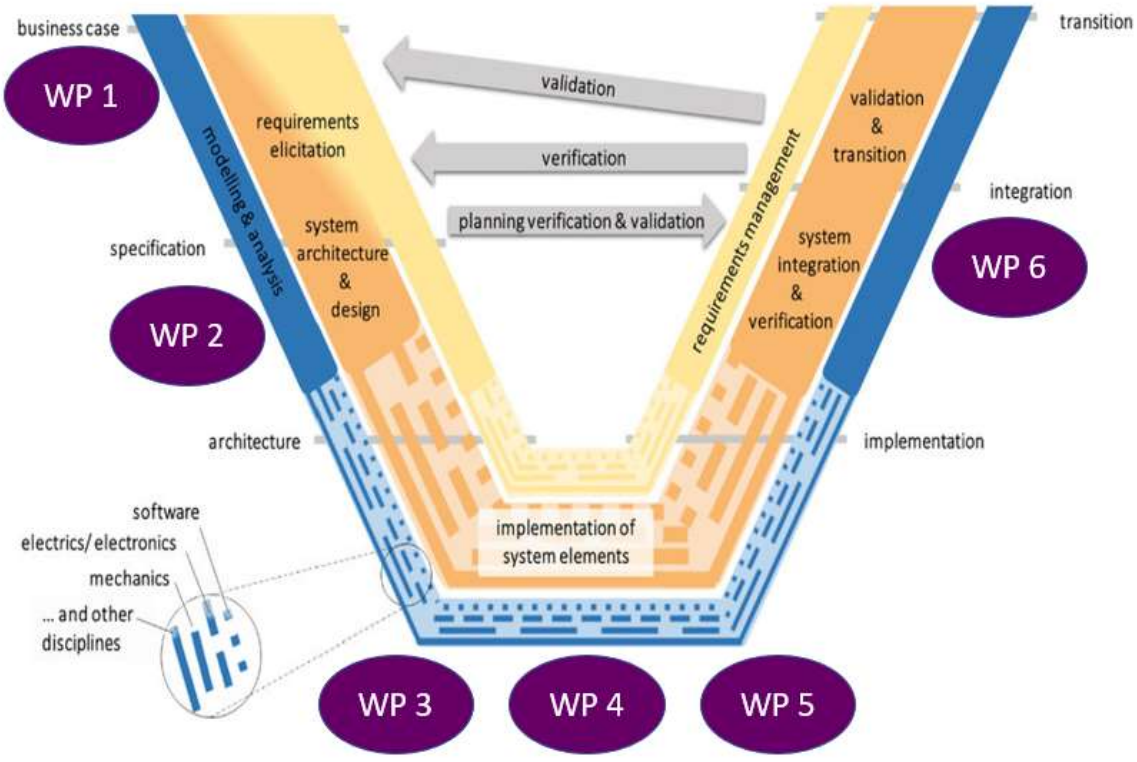


Figure 72. The new V-Model (Graessler, Hentze 2020) with assigned WPs of the i4Q Project

6. Conclusions

Deliverable D1.9 is a Functional Specification document that provides an in-depth definition of the functionalities/behaviours of all i4Q Solutions and explains how related requirements will be fulfilled. D1.9 is the second of two deliverables in the task *Requirements Analysis and Functional Specification*. The requirements engineering process in the i4Q Project that has started in D1.4, is continued in this second deliverable by eliciting, discussing, and refining all requirements. A functional specification process through system modelling has been started by creating Function Structure Diagrams (FSD) for all i4Q Solutions, which serve as a basis for the functional specification and the exact mapping of the requirements to the solutions.

Since the requirements engineering in the i4Q Project is focused on stakeholder, system, and software requirements which all contribute to fulfil the i4Q business requirements, e.g., the realization of Zero-Defect-Manufacturing, a suitable methodology and approach have been developed and presented in Section 2. It is based on standards such as ISO/IEC/IEEE 29148 (ISO/IEC/IEEE 29148:2018-11), ISO/IEC/IEEE 12207 (ISO/IEC/IEEE 12207:2017-11) and ISO/IEC/IEEE 15288 (ISO/IEC/IEEE 15288:2015-05-15), and on the method of Function Based Systems Engineering, FBSE (INCOSE 2017, p. 295ff). These standards have been used as the basis for collecting factories' and system interfaces' requirements for Smart Manufacturing in the context of quality control and they have been adapted to the i4Q approach. Additionally, the process has been performed with respect to the VDI guideline 2221 which describes the design of technical products and systems (VDI 2221-1:2019-11 and VDI 2221-2:2019-11) and the VDI guideline 2206 (VDI 2206:2004-06) which describes the design methodology for mechatronic systems. This guideline VDI2206 has been also used in the current draft version (VDI 2206:2020-09) to describe the further use of the results of this deliverable.

In order to collect all the information and discuss all the requirements and functionalities, the approach of Model-Based Systems Engineering (MBSE) was chosen and requirements and functional diagrams were iteratively created in the open-source system modelling language SysML[32]. The software tools diagrams.net[33] and Cameo system modeler[34] supported this work. The resulting diagrams were used as a communication basis for all partners.

In Section 3 the requirements that were elicited in the first deliverable of this task (D1.4) have been further developed. The pilot requirements have been structured and refined in SysML diagrams, and solution requirements have been elicited through a template. There have been discussions with all partners to identify the specific pilot requirements, to describe them in an understandable way and to cluster them and find the specific i4Q Solutions that will satisfy them. The precise mapping of requirements to the sub-functions of the i4Q Solutions is presented in the diagrams in Section 3 for each pilot case.

[32] www.sysml.org
[33] https://app.diagrams.net/
[34] https://www.3ds.com/products-services/catia/products/no-magic/cameo-systems-modeler/

In Section 4, FSDs have been created for each solution to identify the main functions, the sub-functions, the data flow, as well as the input and output of each solution. These diagrams describe the functionality of each i4Q Solution and have been used for the description of solution interfaces and the corresponding requirements. The precise mapping of requirements to sub-functions is presented for each i4Q Solution. The combination of the functionalities and the mapping of user requirements and solution requirements leads to the functional specifications of all i4Q Solutions. The goal of specifying all system (sub-)functions through requirements could be achieved in many cases. In some cases, it is not possible to precisely describe the functions or even the requirements in the current project phase because some information is not yet available and will only become known in the upcoming work packages. For these cases, the further procedure has been described. Overall, the Functional Specification contains a brief description of the main functions and objectives of each solution, the FSD including the processes and interfaces, the description of the to-be situation through the mapped requirements of users and the interoperability with other solutions and systems in the technical structure with its interfaces. To analyse the strengths and weaknesses of the solutions, the functional specification has been evaluated and explained in their completeness, precision, interface specifications and requirements origins to define the further process in the BUILD work packages.

The evaluation and use of results are described in Section 5. The evaluation of the procedure with the iteration steps and system modelling described in Section 2 has been successfully performed. Interaction with D1.8 has been well coordinated through the use of the pilot requirements diagrams. Interaction with the DESIGN work package 2 was well coordinated through the use of the FSD. All of these diagrams served as a communication basis in the technical discussions and resulted to an overview of the functionalities and requirements of the complete set of i4Q RIDS. The results of the *status comparison* and *req origin comparison* show that overall, there are some solutions that are already quite completely described by requirements, while other i4Q Solutions need to be defined and described in more detail in the upcoming WP. In particular, abstract i4Q solutions such as i4Q^{BC} and i4Q^{AI} received fewer requirements assigned because they do not interact directly with end users. Solutions such as i4Q^{DR} and i4Q^{DIT} received many requirements because they are explicit in their use and the user requirements are also clear. The *solution map* shows the interface connections between all i4Q Solutions. This leads to the further use of the results of this deliverable, which is described in accordance with the new V-model (VDI2206:2020) that contains a complete requirements' engineering process throughout the entire development process. Validation and verification of the requirements fulfilment will be elaborated in WP6 when all solutions will be implemented and evaluated in the pilot factories.

The two perspectives of pilots and solution providers presented in Section 2.1 have been described and validated in this deliverable. All six pilots defined their complete set of user requirements and the solution providers defined the functionalities and interfaces of their solutions as well as corresponding solution interface requirements. The mapping of requirements and functions leads to a clear overview of the Functional Specification of all i4Q Solutions.

References

AS9100, *Quality Systems - Aerospace - Model for Quality Assurance in Design, Development, Production, Installation and Servicing* (https://www.sae.org/standards/content/as9100/).

Bender, B.; Gericke, K. (Hrsg.) (2021). *Pahl/Beitz Konstruktionslehre. Methoden und Anwendung erfolgreicher Produktentwicklung.* Heidelberg: Springer Vieweg.

Cameo System Modeler - https://www.3ds.com/products-services/catia/products/no-magic/cameo-systems-modeler/ (September 2021)

Draw.io - https://app.diagrams.net/ (September 2021)

Graessler, I.; Hentze, J. (2020): *The new V-Model of VDI 2206 and its validation*. In: at - automatisierungstechnik 2020; 68(5), pp. 312-324. DOI: 10.1515/auto-2020-0015

INCOSE. (2015). *INCOSE Systems Engineering Handbuch. Ein Leitfaden für Systemlebenszyklus-Prozesse und -Aktivitäten*. (engl. *INCOSE Systems Engineering Handbook – A Guide for System Life Cycle Processes and Activities*.). NJ: John Wiley & Sons, Inc.

ISO/IEC/IEEE 12207:2017-11, *Systems and software engineering - Software life cycle processes* (ISO/IEC/IEEE 12207:2017-11)

ISO/IEC/IEEE 15288:2015-05-15, *Systems and software engineering - System life cycle processes* (ISO/IEC/IEEE 15288:2015-05-15)

ISO/IEC/ IEEE 29148:2018-11, *Systems and software engineering - Life cycle processes - Requirements engineering* (ISO/IEC/ IEEE 29148:2018-11)

Liu, L.; Chi, L., 2002. *Evolutional Data Quality: A Theory-Specific View*. In *ICIQ,* pp. 292-304.

MBSE - https://mbseworks.com (September 2021)

Nowak-Meitinger, A. M.; Jochem, R. (2021). i4Q – *Deliverable 1.4 - Requirements Analysis and Functional Specification*. (Submitted 29-04-2021).

SysML - www.sysml.org (September 2021)

VDI 2221-1:2019-11, *Design of technical products and systems, Part 1: Model of product design* (VDI 2221-1:2019-11)

VDI 2221-2:2019-11, *Design of technical products and systems, Part 2: Configuration of individual product design processes* (VDI 2221-2:2019-11)

VDI 2206:2004-06, *Design methodology for mechatronic systems* (VDI 2206:2004-06)

VDI 2206:2020-09 (Entwurf), *Entwicklung cyber-physischer mechatronischer Systeme (CPMS) (engl.: Draft, Development of cyber-physical mechatronic systems (CPMS)).* (VDI 2206:2020-09)

Weilkiens, T. (2014). *Systems Engineering mit SysML/UML. Anforderungen, Analyse, Architektur.* 3. Auflage, dpunkt.verlag.

Zafirov, R. (2014). *Modellbildung und Spezifikation*. In: Eigner, M.; Roubanov, D.; Zafirov, R. (Hrsg.): *Modellbasierte Virtuelle Produktentwicklung*. Heidelberg: Springer Vieweg.

Appendix I – List of Solution Requirements

i4Q Solution	ID	Type of Require-ment		Requirement Title	Requirement definition (related to the i4Q Solution)	Require-ment coming from connected Solution	side of connec-ted solution (input/ output)
1-i4Q_DQG Data Quality Guidelines	01BIBAr1	Usability and Quality req		Easy to operationalize	The guideline should be easy to operationalize.	1-i4Q_DQG	
	01BIBAr2	Guidelines req		Cover long-term and short-term measures	The guideline should cover long-term and short-term measures to improve data quality.	1-i4Q_DQG	
	01BIBAr3	Guidelines req		Focus on information	The guideline should focus on information, not on database quality improvements.	1-i4Q_DQG	
	01BIBAr4	Guidelines req		Use data life cycle model	The guideline should use a data life cycle model to define its scope.	1-i4Q_DQG	
2-i4Q_QE QualiExplore for Data Quality Factor Knowledge	02BIBAr1	Functional/ Technical req	Usability and Quality req	Access via website	QualiExplore shall be accessible via a website (it is a web-based solution).	2-i4Q_QE	Both
	02BIBAr2	Functional/ Technical req	Usability and Quality req	Editing environment	QualiExplore users shall be able to create, edit, and delete factor descriptions.	2-i4Q_QE	Input
	02BIBAr4	Functional/ Technical req	Usability and Quality req	Filter functions	QualiExplore users shall be able to filter factors (to reduce cognitive load).	2-i4Q_QE	Both

i4Q Solution	ID	Type of Require-ment		Requirement Title	Requirement definition (related to the i4Q Solution)	Require-ment coming from connected Solution	side of connec-ted solution (input/ output)
	02BIBAr6	Usability and Quality req		Coherent terms	QualiExplore shall use terminology used in other i4Q Solutions (coherence).	2-i4Q_QE	Both
	02BIBAr7	Interface req	Usability and Quality req	Integration	Other i4Q Solutions might integrate QualiExplore as a HTML/Javascript widget (usability/ease of access).	2-i4Q_QE	N/A
	02BIBAr3	Usability and Quality req	Guidelines req	Useful knowledge	QualiExplore factor descriptions shall be intelligible and relevant.	2-i4Q_QE	Both
	02BIBAr5	Guidelines req		Credible knowledge	QualiExplore contents should adopt acknowledged terminology from standards and literature (credibility).	2-i4Q_QE	Both
3-i4Q_BC Blockchain Traceability of Data	03IBMr21	Interface req		Blockchain Usability	Collaborative work is needed to establish data models and interfaces so that other solutions may use this service, which is envisioned as part of the underlying infrastructure. (No specific requirements from additional i4Q Solutions)	3-i4Q_BC, others	
	03IBMr2	Interface req		Exposed interfaces	i4Q_BC should provide interfaces to other solution to enable transaction submissions and queries.	All solutions storing data on the BC solution, e.g., DR, DIT	input
4-i4Q_TN	04ITIr1	Functional/	Interface	Reliable	Reliability, robustness, availability and easy scalability of	4-i4Q_TN	Output

i4Q Solution	ID	Type of Require-ment		Requirement Title	Requirement definition (related to the i4Q Solution)	Require-ment coming from connected Solution	side of connec-ted solution (input/ output)
Trusted Networks with Wireless & Wired Industrial Interfaces		Technical req	req	communications	plant communication networks should allow the network to be able to interconnect a large number of mobile devices.		
	04ITIr2	Functional/ Technical req	Interface req	Low Latency wireless communication	Ultra Reliable and Low Latency Communications (URLLC) should be achieved through the use of synchronized and deterministic mechanism.	4-i4Q_TN	Output
	04ITIr3	Functional/ Technical req	Interface req	Full network orchestration	A dynamic and fully centralize resource provisioning shall be integrated into a WPAN network based in TSCH MAC.	4-i4Q_TN	Input
	04ITIr4	Functional/ Technical req	Interface req	Improve node mobility	Node mobility in WPAN networks shall be improved by using SDN mechanisms to manage data flow changes on the network.	4-i4Q_TN	Input
	04TIAGr1	Functional/ Technical req	Interface req	Traffic convergence	TSN based systems shall allow mixing of time-sensitive and non-time-sensitive traffic on same wire, while guaranteeing that properties of time-sensitive traffic are not affected by the variable load of the non-time sensitive traffic.	4-i4Q_TN	Input/Out put
	04TIAGr2	Functional/ Technical req	Interface req	TSN Support	All devices inside the system that want to use TSN, shall support the TSN protocol. Meaning all switches and end systems shall support TSN functions.	4-i4Q_TN	Input/Out put
	04TIAGr3	Functional/ Technical	Timing Requiremen	Master Clock	Each end device and switch shall be synchronisable with a grand master clock	4-i4Q_TN	Input/Out put

i4Q Solution	ID	Type of Require-ment		Requirement Title	Requirement definition (related to the i4Q Solution)	Require-ment coming from connected Solution	side of connec-ted solution (input/ output)
		req	t				
	04TIAGr4	Functional/ Technical req	Timing Requiremen t	Bandwidth	A maximum bandwidth available shall be 10Gbit/sec	4-i4Q_TN	Input/outp ut
	04IKERr1	Functional/ Technical req	Interface req	Protocol translation	Translation between fieldbus network (CAN, MODBUS, ...) and IoT protocol (MQTT, CoAP, AMQP, ...) shall be provided.	6-i4Q_SH	-
5-i4Q_CSG Cybersecurit y Guidelines	05IKERr1	Guidelines req		Provide trust mechanisms	The cybersecurity guidelines shall provide trust mechanism based on IEC 62443 standard.	5-i4Q_CSG	
	05IKERr2	Guidelines req		Describe security mechanisms	The cybersecurity guidelines should describe security mechanism both for an IACS (Industrial Automation & Control Systems) topology and for individual components.	5-i4Q_CSG	
6-i4Q_SH IIoT Security Handler	06IKERr1	Functional/ Technical req	Security req	Provide trust (x509 certificates)	When new i4Q module is set up, the i4Q_SH shall provide trust in terms of a digital identity with x509 certificates.	6-i4Q_SH	
	06IKERr2	Functional/ Technical req	Security req	Digital identity storage	An i4Q module shall store the digital identity in a secure way. If the i4Q module cannot store the digital identity it should use the cryptography operations of the i4Q_SH.	6-i4Q_SH	
	06IKERr3	Functional/ Technical	Security req	Communication ciphering among	When an i4Q module communicates with another i4Q module it should cipher the communication using the	6-i4Q_SH	

i4Q Solution	ID	Type of Require-ment		Requirement Title	Requirement definition (related to the i4Q Solution)	Require-ment coming from connected Solution	side of connec-ted solution (input/ output)
		req		modules	provided trust.		
	06IKERr4	Functional/ Technical req	Security req	Communication ciphering in storage	When a i4Q module stores data in a secure way it shall use the provided trust to cipher the communication.	6-i4Q_SH	
	06IKERr5	Functional/ Technical req	Security req	Provide cryptography operations	i4Q_SH shall provide cryptography operations supported by one HSM.	6-i4Q_SH	
7-i4Q_DRG Data Repository Guidelines	07ITIr1	Guidelines req	Usability and Quality req	Describe repository use	A technical document describing the whole process of the repository creation shall be provided. It will be as intuitive and complete as possible, to serve as a guide for future implementations. It will take some input from i4Q_DR.	7-i4Q_DRG, 8-i4Q_DR	
	PC3r3.3	Functional /Tehnical req.		All data generated by new modules shall be adhering to provided ontology MPFQ (developed in H2020-QU4LITY) Evaluate and embed MPFQ model into i4Q	The existing MPFQ (Material-Process-Functions-Quality) Model should be evaluated and embedded, and the data should be mapped (harmonized) on it	7-i4Q_DRG	

i4Q Solution	ID	Type of Require-ment		Requirement Title	Requirement definition (related to the i4Q Solution)	Require-ment coming from connected Solution	side of connec-ted solution (input/ output)
				to harmonize the data.			
8-i4Q_DR Data Repository	08ITIr1	Functional/ Technical req	Usability and Quality req	Performant data storage	The desired performance levels shall be achieved, being able to process the incoming data and requests in a reasonable time. To do so, some factors as scalability or even outsourcing some processing tasks to public clouds shall be considered.	8-i4Q_DR	Output
	08ITIr2	Functional/ Technical req	Interface req	Complete data storage	The repository shall be able to hold all the necessary information and to store the full size of the incoming data.	8-i4Q_DR	Output
	08ITIr3	Interface req		Accessible data storage	The repository shall communicate successfully with the other components of the system and provide the necessary channels and interfaces to make the information transmission feasible.	8-i4Q_DR	Input, Output
	08ITIr4	Functional/ Technical req		Optimized data storage	The repository shall store the data in an optimized way. A data model shall be designed to organize all the information suitably (from the volume, data processing, machine learning, etc. points of view).	8-i4Q_DR	Output
	08ITIr5	Security req		Secure data storage	The necessary security mechanisms shall be implemented to ensure the protection of private data (encryption) and the human access to it (authentication, authorization).	8-i4Q_DR	Input, Output
	08TUBr1	Functional/		Simple way to	The connection to the data repository should be	17-i4Q_PQ	Input

i4Q Solution	ID	Type of Require-ment		Requirement Title	Requirement definition (related to the i4Q Solution)	Require-ment coming from connected Solution	side of connec-ted solution (input/ output)
		Technical req		connect data repository in scripts for implementation purposes	established in a common way to have a direct data pipeline to "stream" data directly to the solution		
	08TUBr2	Functional/ Technical req		Storage of analytical results	The data repository should be able to store analytical results over long term via writing directly into the data repository (or any other easy way)	17-i4Q_PQ	Output
	08IKERr1	Functional/ Technical req		Storage of DT models	The data repository should be able to store models developed inside the DT solution (path to the model location), together with its metadata (defining the characteristics of the model)	16-i4Q_DT 19-i4Q_PA	Input/Output Input
	08UNIr1	Functional/ Technical req		Simple way to connect to data	The connection to the data repository should be established in a common way to have a direct data pipeline to batch data directly to the solution as well as direct connection to streams of data	10-i4Q_DA 11-i4Q_BDA 12-i4Q_AD	Input
	08UNIr2	Functional/ Technical req		Storage of analytical results and/or models	The data repository should be able to store analytic models for future use as well the results coming from the analytic models	10-i4Q_DA 11-i4Q_BDA	Output
	08CERTHr1	Functional/ Technical req		Ability to write data	User access should be granted to write data in the repository	09-i4Q_DIT	Output

i4Q Solution	ID	Type of Require-ment		Requirement Title	Requirement definition (related to the i4Q Solution)	Require-ment coming from connected Solution	side of connec-ted solution (input/ output)
	08CERTH r2	Functional/ Technical req		Ability to read data	User access should be granted to read data from the repository	09-i4Q_DIT	Input
	08CERTH r3	Functional/ Technical req		Connect the data repository for model training and validation purposes	The data repository should provide data to train and validate the predictive algorithms in order to extract information about faults.	15-i4Q_IM, 18-i4Q_QD	Input
	08CERTH r4	Functional/ Technical req		Store the results of the analysis	The analytical data resulting from the Rapid Quality Diagnosis solution should be able to connect to the Data Repository for long term storage.	18-i4Q_QD	Output
	08IKERr2	Functional/ Technical req		Store the results from DT	The simulation results from DT should be stored in the DR so that other solutions can exploit them	16-i4Q_DT	Output
	08UPVr1	Functional/ Technical req		Write and read data	It shall be possible to provide automated access for different applications to write and read data in the repository.	21-i4Q_LRT	Input and Output
	08FARPr 1	Functional/ Technical req		Image data recording	The stored image data should be recorded during the defect detection process and offers production managers, but also customers, the possibility of real-time monitoring.		
9-i4Q_DIT	09CERTH	Functional/		Draw data from	Data from the data repository/ pilots should be drawn.	8-i4Q_DR	input

i4Q Solution	ID	Type of Require-ment		Requirement Title	Requirement definition (related to the i4Q Solution)	Require-ment coming from connected Solution	side of connec-ted solution (input/ output)
Data Integration and Transformation	r1	Technical req		repositories			
	09CERTH r2	Interface req		Connection with microservices	This solution should be able to connect with other microservices.	9-i4Q_DIT	input/output
	09CERTH r3	Functional/ Technical req		Data preparation for pre-processing	Data preparation actions on input data shall be performed.	9-i4Q_DIT	input
	09CERTH r4	Functional/ Technical req		Data integration depending on data	Group integration actions shall be performed according to the nature of data that will be combined.	9-i4Q_DIT	output
	09CERTH r5	Functional/ Technical req		Data fusion	Early fusion of input data should be performed.	9-i4Q_DIT	
	09UNIr1	Functional/ Technical req		Simple way to connect to manufacturing data streams	The connection to the data integration should be established to have a direct data pipeline to manufacturing data streams	10-i4Q_DA 11-i4Q_BDA 12-i4Q_AD	input
	09CERTH r6	Functional/ Technical req		Connect 9-i4Q_DIT solution for model training and validation, and	The data integration solution should provide data to train the predictive algorithms in order to extract information about faults.	15-i4Q_IM, 18-i4Q_QD	Input

i4Q Solution	ID	Type of Require-ment		Requirement Title	Requirement definition (related to the i4Q Solution)	Require-ment coming from connected Solution	side of connec-ted solution (input/ output)
				evaluation			
	09UPVr1	Functional/ Technical req		Connection with microservices	We must be able to connect to a client to generate continuous communication with the possibility to do read/write and publish/subscribe.	21-i4Q_LRT	Input/outp ut
	09AIMPr1	Functional/ Technical req		Synchronize image data	The i4Q Solution should synchronise all data in the process, the process data of the injection moulding machine and any additional sensors and correlate them with the artificial image data.		
10-i4Q_DA Services for Data Analytics	10UNIr1	Usability and Quality req		User access	A user access to Data Analytics Services shall be provided: (Registration and login/ download of a containerized deployment bundle).	10-i4Q_DA	
	10UNIr2	Interface req		Data connection	Data access shall be provided: the i4Q Data Repository or other types of repositories should be accessible/data connectors should be available to connect selected data sources.	10-i4Q_DA	
	10UNIr3	Usability and Quality req		Data selection by the User	Data selection functionality should be provided (user selects input data for the desired service).	10-i4Q_DA	
	10UNIr4	Security req		Data security	Security: Data storage and exchange should be secured. (non-functional)	10-i4Q_DA	
	10UNIr5	Usability and Quality		Data scalability	Scalability to 10s of nodes: Solution should be scalable to any type of environment (cloud, on-premises). (non-	10-i4Q_DA	

i4Q Solution	ID	Type of Require-ment		Requirement Title	Requirement definition (related to the i4Q Solution)	Require-ment coming from connected Solution	side of connec-ted solution (input/ output)
		req			functional)		
	10UNIr6	Functional/ Technical req	Usability and Quality req	Results provision	Results shall be provided: provide the results through web service/bundle or in the i4Q_AD.	10-i4Q_DA, 12-i4Q_AD	
	10TUBr1	Functional/ Technical req		Easy to use analytical framework integration	Depending on the specific framework which is used for deploying the solution, a framework description including a wiki or a handbook should be provided.	17-i4Q_PQ	Input
11-i4Q_BDA Big Data Analytics Suite	11UNIr1	Interface req		User access	User access to i4Q_BDA should be provided (Registration and login/Download of containerized deployment Bundle).	11-i4Q_BDA	
	11UNIr2	Functional/ Technical req	Interface req	Bundle configuration	The user of i4Q_BDA should be able to define desired tools, methods, libraries and technologies, as well as the target deployment infrastructure Cloud/Edge. This should be fully coordinated with i4Q_AI.	11-i4Q_BDA, 13-i4Q_AI	
	11UNIr3	Functional/ Technical req	Usability and Quality req	Customization interface	i4Q_BDA should provide an interface that contains the tools/methods required to provide the best solution.	11-i4Q_BDA	
	11UNIr4	Security req		Data security	Security: Data storage and exchange should be secured. (non-functional)	11-i4Q_BDA	
	11UNIr5	Interface		Data connection	Data access: the i4Q_DR or other types of repositories should be accessible/data connectors should be available	11-i4Q_BDA	

i4Q Solution	ID	Type of Require-ment		Requirement Title	Requirement definition (related to the i4Q Solution)	Require-ment coming from connected Solution	side of connec-ted solution (input/ output)
		req			to connect selected data sources.		
	11UNIr6	Functional/ Technical req		API methods provision	The solution should provide an open API with all the methods and algorithms available.	11-i4Q_BDA	
	11UNIr7	Usability and Quality req		Data scalability	Scalability to 10s of nodes: Solution should be scalable to any type of environment (cloud, on-premises). (non-functional)	11-i4Q_BDA	
	11TUBr1	Functional/ Technical req		Easy to use analytical framework integration	Depending on the specific framework which is used for deploying the solution, a framework description including a wiki or a handbook should be provided.	17-i4Q_PQ	Input
12-i4Q_AD Analytics Dashboard	12UNIr1	Usability and Quality req		User access	User access to Analytics Dashboard should be provided (Registration and login/ download of a containerized deployment bundle).	12-i4Q_AD	
	12UNIr2	Interface req		Data connection	Data access: the i4Q_DR or other types of repositories should be accessible/data connectors should be available to connect selected data sources.	12-i4Q_AD	
	12UNIr3	Usability and Quality req		Data selection by User	Configurable Data selection functionality should be integrated (user selects data for visualization).	12-i4Q_AD	
	12UNIr4	Security req		Data security	Security: Data storage and exchange should be secured.	12-i4Q_AD	

i4Q Solution	ID	Type of Require-ment		Requirement Title	Requirement definition (related to the i4Q Solution)	Require-ment coming from connected Solution	side of connec-ted solution (input/ output)
					(non-functional)		
	12UNIr5	Functional/ Technical req	Interface req	Results Provision	Results of i4Q_DA and i4Q_BDA shall be provided and presented in i4Q_AD (connection to i4Q_DA and i4Q_BDA).	10-i4Q_DA, 11-i4Q_BDA, 12-i4Q_AD	
	12TUBr1	Functional/ Technical req		Visualize Process Quality Data	Stability, capability and performance (quality output) of manufacturing processes shall be visualized in a dashboard solution (Code is provided by TUB).	17-i4Q_PQ	Output
	12TUBr2	Functional/ Technical req		Capability to read Python script	Dashboard should be compatible with python libraries to display the visualization.	17-i4Q_PQ	Output
13-i4Q_AI AI Models Distribution to the Edge	13IBMr1	Functional/ Technical req	Interface req	Interface with different solutions (QD)	i4Q_AI, i4Q_EW, and i4Q_IM shall be fully coordinated to create a consistent platform to service the data analytics services such as i4Q_PQ and i4Q_QD.	13-i4Q_AI, 14-i4Q_EW, 15-i4Q_IM	Input
	13IKERr1	Functional/ Technical req	Interface req	Interface with different solutions (DT)	Since multiple simulations will need to be run (either in edge or in cloud), it would be beneficial that i4Q_EW, i4Q_AI, i4Q_DA should have the ability to handle i4Q_DT simulations.	16-i4Q_DT	Input
	13UNIr1	Functional/ Technical req	Interface req	Model workload Distribution	The AI models distribution to the Edge should handle the distribution of the analytical model's workload through the edge nodes when needed	10-i4Q_DA, 11-i4Q_BDA	Input
14-i4Q_EW Workloads	14UNIr1	Functional/ Technical	Interface	Interface with different	Bundle configuration: the user of i4Q_BDA should be able to define desired tools, methods, libraries and	11-i4Q_BDA, 13-i4Q_AI,	Input

i4Q Solution	ID	Type of Require-ment		Requirement Title	Requirement definition (related to the i4Q Solution)	Require-ment coming from connected Solution	side of connec-ted solution (input/ output)
Placement and Deployment		req	req	solutions (BDA)	technologies, as well as the target deployment infrastructure Cloud/Edge. This should be fully coordinated with i4Q_AI and i4Q_EW.	14-i4Q_EW	
	14IBMr1	Functional/ Technical req	Interface req	Interface with different solutions (QD)	i4Q_AI, i4Q_EW, and i4Q_IM shall be fully coordinated to create a consistent platform to service the data analytics services such as i4Q_PQ i4Q_QD.	13-i4Q_AI, 14-i4Q_EW, 15-i4Q_IM	Input
	14IKERr1	Functional/ Technical req	Interface req	Interface with different solutions (DT)	Since multiple simulations will need to be run (either in edge or in cloud), it would be beneficial that i4Q_EW, i4Q_AI, i4Q_DA should have the ability to handle i4Q_DT simulations.	16-i4Q_DT	Input
15-i4Q_IM Infrastructur e Monitoring	15IBMr1	Functional/ Technical req	Interface req	Monitor other analytic solutions	i4Q_AI, i4Q_EW, and i4Q_IM shall be fully coordinated to create a consistent platform to service the data analytics services such as i4Q_PQ and i4Q_QD.	13-i4Q_AI, 14-i4Q_EW, 15-i4Q_IM	
	15AIMPr1	Functional/ Technical req		Monitor as a standalone	Relevant data should be collected in order to have the necessary information in order to agile the decision making, and correlate this data with quality data to identify quickly the problem in order to solve it.	15-i4Q_IM	
	15CERTH r1	Functional/ Technical req		Connect data from the manufacturing asset repository.	Historical data from sensors should be available in order to build the solution.	15-i4Q_IM	
	15CERTH	Functional/		Generate alerts	The solution should be able to manage and provide	15-i4Q_IM	

i4Q Solution	ID	Type of Require-ment		Requirement Title	Requirement definition (related to the i4Q Solution)	Require-ment coming from connected Solution	side of connec-ted solution (input/ output)
	r2	Technical req			alerts.		
	15CERTH r3	Interface req		Send alerts to user	The solution should be able to access the edge environment.	15-i4Q_IM	
	15CERTH r4	Functional/ Technical req		Failure detection	The solution should successfully detect failures in the processes it monitors.	15-i4Q_IM	
16-i4Q_DT Digital Twin simulation services	16IKERr1	Functional/ Technical req		Flexible UI to develop different models	Flexible enough Digital Twin model template (UI) with the minimum possible dependencies on the type of input data and manufacturing/asset plant shall be developed.	16-i4Q_DT	
	16IKERr2	Functional/ Technical req		Flexible output visualization	The output visualization of the Digital Twin should be depicted in a general way to be able to provide simulation results to other components in the least constrained way possible.	16-i4Q_DT	
	16CERTH r1	Functional/ Technical req		Create train and validation datasets	The Digital Twin simulation services should provide data to train and validate the models.	18-i4Q_QD	Input
	16UPVr1	Functional/ Technical req	Interface req	Send different configurable data	When using DT by the LRT solution, it needs to be able to send different configurable data for its correct use.	21-i4Q_LRT	Input
	16UNIr1	Functional/ Technical		Simple way to connect to	The connection to the data integration should be established to have a direct data pipeline to	10-i4Q_DA 11-i4Q_BDA	input

i4Q Solution	ID	Type of Require-ment		Requirement Title	Requirement definition (related to the i4Q Solution)	Require-ment coming from connected Solution	side of connec-ted solution (input/ output)
		req		manufacturing data streams	manufacturing data streams	12-i4Q_AD	
17-i4Q_PQ Data-driven Continuous Process Qualification	17TUBr1	Functional/ Technical req		Reduce ramp-up time	The time to process approval during ramp-up phase shall be reduced through stable processes with available process data acquired from simulated data or predecessor process, full access to process related data sources, sufficient data quality and quantity is needed for algorithm modelling.	17-i4Q_PQ	
	17TUBr2	Functional/ Technical req	Interface req	Continuously monitor process capability	In-line continuous process validation after process reconfiguration should be possible through data-driven continuous process qualification, full access to process related data sources, sufficient data quality and quantity is needed for algorithm modelling.	17-i4Q_PQ	
	17TUBr3	Functional/ Technical req	Usability and Quality req	Visualization of Process performance	Stability, capability and performance (quality output) of manufacturing processes shall be monitored just-in-time and visualized in a dashboard solution.	17-i4Q_PQ	
	17FARPr 1	Functional/ Technical req		Automated defect detection	Automated defect detection system to eliminate defects on the visual side of the part. System should give same performance for every production shift on every environmental condition (day-light, vibration, etc).		
18-i4Q_QD Rapid	18CERTH r1	Functional/ Technical		Train different models	This microservice should have different modules for each function it performs, e.g., detection, decision and decision	18-i4Q_QD	

i4Q Solution	ID	Type of Require-ment		Requirement Title	Requirement definition (related to the i4Q Solution)	Require-ment coming from connected Solution	side of connec-ted solution (input/ output)
Quality Diagnosis		req			making.		
	18CERTHr2	Interface req		Interaction with other i4Q Solutions	The solution should interact with other solutions for monitoring and receiving input.	18-i4Q_QD	
	18CERTHr3	Functional/ Technical req		Store the results of the analysis	The solution should store the information it receives.	18-i4Q_QD	
	18CERTHr4	Functional/ Technical req		Visualize Data	This microservice should be able to produce KPI's.	18-i4Q_QD	
19-i4Q_PA Prescriptive Analysis Tools	19IKERr1	Interface req		Flexible interaction with DT	An efficient and flexible enough communication with other i4Q Solutions, particularly with i4Q_DT should be ensured.	19-i4Q_PA	
	19IKERr2	Functional/ Technical req	Usability and Quality req	Interface for scenarios and evaluation criteria definition	An interface should be provided that allows the definition of the scenarios to be simulated and the criteria to evaluate the simulation results and provide a prescriptive analysis.	19-i4Q_PA	
	19IKERr3	Functional/ Technical req	Interface req	Allow running multiple simulations	Since multiple simulations will need to be run (either in edge or in cloud), it would be beneficial that i4Q_EW, i4Q_AI, i4Q_DA should have the ability to handle i4Q_DT simulations.	16-i4Q_DT	Output

i4Q Solution	ID	Type of Require-ment		Requirement Title	Requirement definition (related to the i4Q Solution)	Require-ment coming from connected Solution	side of connec-ted solution (input/ output)
	19UNIr1	Functional/ Technical req		Simple way to connect to manufacturing data streams	The connection to the data integration should be established to have a direct data pipeline to manufacturing data streams	10-i4Q_DA 11-i4Q_BDA 12-i4Q_AD	input
20-i4Q_LRG Manufacturing Line Reconfiguration Guidelines	20UPVr1	Guidelines req	Usability and Quality req	User guide about Manufacturing Line reconfiguration	An interactive and intuitive guide for users about Manufacturing Line reconfiguration should be provided. (No specific requirements from additional i4Q Solutions.)	20-i4Q_LRG	
	20UPVr2	Guidelines req	Usability and Quality req	Technical document	A technical document of the manufacturing line (it will be intuitive) should be provided.	20-i4Q_LRG	
	20UPVr3	Functional/ Technical req	Usability and Quality req	Store and facilitate historical query data	This solution should store historical query data to facilitate its use to the end-user.	20-i4Q_LRG	
21-i4Q_LRT Manufacturing Line Reconfiguration Toolkit	21UPVr1	Functional/ Technical req		Collect relevant data	Relevant data shall be collected in order to have the necessary information to speed up the decision-making process of the production line, considering quality.	21-i4Q_LRT	
	21UPVr2	Interface req		Established connectivity	The toolkit should be able to connect with other i4Q Solutions and communicate successfully with the other components of the system.	21-i4Q_LRT	
	21UPVr3	Functional/		Generic database	A database of the production line data shall be collected	21-i4Q_LRT	

i4Q Solution	ID	Type of Require-ment		Requirement Title	Requirement definition (related to the i4Q Solution)	Require-ment coming from connected Solution	side of connec-ted solution (input/ output)
		Technical req			from other i4Q Solution and maintained in order to train the AI.		
	21UPVr4	Functional/ Technical req		Detailed analysis	A study of the machinery, cell or manufacturing line shall be carried out in order to find out the points of analysis.	21-i4Q_LRT	
	21UPVr7	Functional/ Technical req		Client Management	LRT shall be able to connect to the different clients to be able to use this solution (streaming services, file storage services, databases). These clients must be directly the machine or services that use the solution. In this way to be able to visualise or use the results of the algorithms launched.	21-i4Q_LRT	
	21UPVr8	Functional/ Technical req		Distributable file	LRT shall provide a distributable file that can be launched on any server at the edge.	21-i4Q_LRT	
	21FARPr1	Functional/ Technical req		Production parameter optimization	Part quality prediction algorithms should be introduced to optimize production parameters accordingly.		
22-i4Q_LCP Manufacturing Line Data Certification Procedure	22TUBr1	Guidelines req		Develop Audit Procedure	An audit procedure for manufacturing resources (machine, cell or manufacturing line) shall be developed to ensure the accuracy and reliability of data from the manufacturing process.	22-i4Q_LCP	
	22TUBr2	Guidelines	Usability	Knowledge	Knowledge shall be transferred to the employees through	22-i4Q_LCP	

i4Q Solution	ID	Type of Require-ment		Requirement Title	Requirement definition (related to the i4Q Solution)	Require-ment coming from connected Solution	side of connec-ted solution (input/ output)
		req	and Quality req	Management	training according to guidelines and applied standards.		
All i4Q Solutions (RIDS)	GEN_TUBr1	General Req		Digital process transformation	Data analytics and data-driven approaches should support complete digital transformation of process related data.		
	GEN_TUBr2	General Req		Traceability of defect	Automatic Traceability of defects should be clear, consistent and continuous labelling of all components and products throughout the value-added process.		
	GEN_TUBr3	General Req		Data Acquisition, process, storage and access	Industrial partners should use data acquisition, processing, storage and accessing technologies.		
	GEN_IBMr1	General Req		Adaptability of edge computing platform	The development of edge computing platform should be addressing a broad set of current and future requirements from industrial and manufacturing use cases. The edge platform is expected to address a wide range of aspects as defined by i4Q. IBM's contributions will address aspects of enabling intelligent data processing at the manufacturing floor by bringing AI/ML to the edge close to where the data is being generated. Having these capabilities at the edge will significantly improve the quality, reliability, and efficiency in manufacturing.		
	GEN_IBM	General Req		Trust and traceability	The development of blockchain (BC) based infrastructure should be addressing data trust and traceability in		

i4Q Solution	ID	Type of Require-ment		Requirement Title	Requirement definition (related to the i4Q Solution)	Require-ment coming from connected Solution	side of connec-ted solution (input/ output)
	r2			Blockchain infrastructure	industrial and manufacturing use cases. The BC platform is expected to address cornerstone issues of data reliability to enable smart processing by analysis components, as well as establishing trust among partners. Having these capabilities will significantly enhance the quality, reliability, and trust in manufacturing data.		
	GEN_AIM Pr1	General Req		Adaptability of i4Q Solutions	The i4Q Solution should be adapted to a variety of different machine types (injection machine makers, year of manufacturing, format of the data process to collect...).		
	GEN_ITIr 1	General Req		Improve data exchange	Improve the reliability and quality of data exchange in the plant by deploying a new generation of private infrastruc-ture based on 5G networks. These networks should offer a highly scalable and independent architecture that increases traffic capacity and network efficiency compared to 4th generation technologies.		
	GEN_ITIr 2	General Req		Improvement in communication latency	Improvement in communication latency should be achieved through various techniques and standardised mechanisms, such as TSN, to achieve wireless communication have a latency similar to wired communications.		
	GEN_ITIr 3	General Req		Introduce WPAN environment	Integrate the software-defined networking paradigm into a WPAN environment with TSCH MAC. This allows to		

i4Q Solution	ID	Type of Require-ment		Requirement Title	Requirement definition (related to the i4Q Solution)	Require-ment coming from connected Solution	side of connec-ted solution (input/ output)
					implement dynamic and fully centralized resource provisioning, increasing flexibility and scalability as well as allowing node mobility while reducing management complexity.		
	GEN_TIA Gr1	General Req		Broad integration of IEE 802.1 TSN	IEE 802.1 TSN (deterministic wired communication) should be integrated into a wide collection of machines and in-crease the use of deterministic communication into industry.		

Table 8. List of Solution Requirements

Appendix II – Requirements Mapping matrices of i4Q^{TN} and i4Q^{DR}

i4Q^{TN}	*PC2r16 i4Q - Data transfer*	*PC4r1.2 Ensure reliability*	*PC6r1.3 Capability of securely moving data*	*4IKERr1 TSN Support*	*4ITIr1 Full network orchestration*	*4ITIr2 Improve node mobility*	*4ITIr3 Low Latency wireless communication*	*4ITIr4 Reliable communications*	*4TIAGr1 Bandwidth*	*4TIAGr2 Master Clock*	*4TIAGr3 Protocol translation*	*4TIAGr4 Traffic convergence*
Solution itself (no concrete mapping)		X	X		X							
Assign network resources												
Check resource availability												
Compare requested resources												
Manage resources												
Monitor and orchestrate network resources via SDN Controller												
Provide network connectivity with Industrial Wireless Sensor Networks	X	X	X			X						
Provide network connectivity with Private 5G technology	X	X	X				X					
Provide network connectivity with TSN technology	X	X	X	X				X	X	X	X	X
Publish Network statistics and new configurations												
Receive Quality Feedback from different networks												
Receive SDN controller commands_IWSN												
Receive SDN controller commands_5G												
Receive SDN controller commands_TSN												
Send control messages_IWSN	X	X	X			X						
Send control messages_5G	X	X	X				X					
Send control messages_TSN	X	X	X					X	X	X	X	X
Translate the controller specification in 5G configuration												
Translate the controller specification in generic network configuration												
Translate the controller specification in IWSN configuration												
Translate the controller specification in TSN configuration												
Translate to network specific management configuration_IWSN												
Translate to network specific management configuration_5G												
Translate to network specific management configuration_TSN												
Update network resource allocation_IWSN	X	X	X			X						

Update network resource allocation_5G	X	X	X	X						
Update network resource allocation_TSN	X	X	X		X	X	X	X	X	
Update network topology										

Table 9. Requirements Mapping matrix of i4Q[TN]

i4Q[DR]	PC1r2.4	PC1r3.2	PC2r3	PC3r3.1	PC3r3.2	PC4r1.1.1	PC4r1.1.2	PC4r1.1.3	PC4r1.3	PC4r4.2	PC4r4.1	PC4r5.1.1	PC4r5.3	PC4r5.4.1	PC5r3.3	PC6r1.4.1	PC6r1.4.2)	PC6r1.4.3)	PC6r8.1	PC6r8.2	PC6r8.3
Solution itself (no concrete mapping)																					
access control																					
check authorization																					
config access control																					
create data store replica												X									
delete image blob																					
delete structured data																					
export to db											X										
get caller info																					
get required permissions																					
import from db							X				X			X							
manage data repository																				X	
manage image blobs										X											
manage structured data										X											
query		X									X			X							
register actions to authorize																					
register authorize bind																					
register authorize rule																					
register available roles																					
remove data store replica																					
save image blob	X			X	X	X	X	X	X		X	X	X		X	X	X	X	X	X	X
save structured data	X		X	X	X	X	X	X	X		X	X	X		X	X	X	X	X		
update image blob											X										X
update structured data											X										

Table 10. Requirements Mapping matrix of i4Q[DR] Part 1

i4Q[DR]	*8CERTHr1*	*8CERTHr2*	*8CERTHr3*	*8CERTHr4*	*8FARPr3*	*8IKERr1*	*8IKERr2*	*8ITIr1*	*8ITIr2*	*8ITIr3*	*8ITIr4*	*8ITIr5*	*8TUBr1*	*8TUBr2*	*8UNIr1*	*8UNIr2*	*8UPVr1*
Solution itself (no concrete mapping)													X		X		
access control	X	X										X					X
check authorization											X						
config access control																	
create data store replica								X									
delete image blob																	
delete structured data																	
export to db												X			X		
get caller info									X								
get required permissions																	
import from db																	
manage data repository								X									
manage image blobs								X									
manage structured data								X									
query			X							X							X
register actions to authorize																	
register authorize bind																	
register authorize rule																	
register available roles																	
remove data store replica																	
save image blob			X	X	X	X			X	X	X			X		X	X
save structured data			X		X	X			X	X	X			X		X	X
update image blob										X							
update structured data										X							

Table 11. Requirements Mapping matrix of i4Q[DR] Part 2

Appendix III – Analysis of Functional Specification Results

	Completeness			Precision			Absolute Interfaces			Req Origin
	# mapped low level System Functions	# all low-level System Functions	**Result**	# mapped low level Requirements	# all mapped Requirements	**Result**	FSD Interfaces (Input)	FSD Interfaces (Output)	**Result**	**Result**
2-i4Q_QE	5	12	**0.42**	4	9	**0.44**	0	0	**0**	**3**
3-i4Q_BC	0	5	**0.00**	0	2	**0.00**	2	0	**2**	**1**
4-i4Q_TN	6	20	**0.30**	9	12	**0.75**	15	3	**15**	**6**
6-i4Q_SH	5	12	**0.42**	3	8	**0.38**	15	15	**15**	**4**
8-i4Q_DR	7	18	**0.39**	31	39	**0.79**	4	8	**12**	**12**
9-i4Q_DIT	7	9	**0.78**	27	32	**0.84**	1	1	**2**	**10**
10-i4Q_DA	5	13	**0.38**	12	20	**0.60**	2	3	**5**	**6**
11-i4Q_BDA	7	11	**0.64**	7	29	**0.24**	2	0	**2**	**6**
12-i4Q_AD	6	10	**0.60**	14	26	**0.54**	7	0	**7**	**8**
13-i4Q_AI	2	6	**0.33**	2	3	**0.67**	2	1	**3**	**3**
14-i4Q_EW	2	6	**0.33**	3	4	**0.75**	1	1	**2**	**4**
15-i4Q_IM	7	9	**0.78**	14	16	**0.88**	2	0	**2**	**8**
16-i4Q_DT	14	24	**0.58**	9	11	**0.82**	3	5	**8**	**7**
17-i4Q_PQ	2	20	**0.10**	4	12	**0.33**	4	2	**6**	**5**
18-i4Q_QD	4	6	**0.67**	13	19	**0.68**	3	1	**4**	**5**
19-i4Q_PA	6	11	**0.55**	6	8	**0.75**	3	2	**5**	**3**
21-i4Q_LRT	3	15	**0.20**	3	14	**0.21**	4	0	**4**	**5**
22-i4Q_LCP	0	9	**0.00**	0	3	**0.00**	1	1	**2**	**2**

Table 12. Analysis of Functional Specification Results