PHYSICS

oPTIMIZED HYbrid Space-time servIce Continuum in faaS

D6.4 – Application Scenarios Definition V2

Lead Beneficiary	**FTDS**
Work Package Ref.	WP6 – Use Cases Adaptation, Experimentation, Evaluation
Task Ref.	T6.2 – Use Cases Scenarios
Deliverable Title	D6.4 – Application Scenarios Definition V2
Due Date	2022-11-30
Delivered Date	2022-12-02
Revision Number	5.0
Dissemination Level	Public (PU)
Type	Other (ORDP)
Document Status	Reviewed
Review Status	Internally Reviewed and Quality Assurance Reviewed
Document Acceptance	WP Leader Accepted and Coordinator Accepted
EC Project Officer	Mr. Stefano Foglietta

H2020 ICT 40 2020 Research and Innovation Action

This project has received funding from the European Union's horizon 2020 research and innovation programme under grant agreement no 101017047

CONTRIBUTING PARTNERS

Partner Acronym	Role	Name Surname
FTDS	Dissemination Leader	Franke, Niklas
FTDS	Dissemination Leader	Hennecke, André
DFKI	Contribution	Herged, Arnold
DFKI	Contribution	Harms, Carsten
ISPRINT	Contribution	op den Akker, Harm
ISPRINT	Contribution	Pnevmatikakis, Aristodemos
ISPRINT	Contribution	Labropoulos, George
CYBE	Contribution	Lohier, Théophile
UPM	Review	Azqueta, Ainhoa
BYTE	Review	Poulakis, Yannis
INNOV	Quality Assurance	Ariana Polyviou

REVISION HISTORY

Version	Date	Partner(s)	Description
0.1	2021-02-12	FTDS	ToC
0.2	2021-04-21	FTDS	First methodology description
0.3	2021-05-06	FTDS	Summary and Methodology description drafted
0.4	2021-05-12	FTDS	Summary and Methodology description reworked
0.5	2021-06-07	FTDS	Questionnaire Description
0.6	2021-06-21	FTDS	Questionnaire and Workshop description
0.7	2021-07-05	DFKI, CYBE, iSPRINT	Input of all three workshops implemented
0.8	2021-07-11	FTDS	Internal Review Version
0.9	2021-07-12	iSPRINT	Completed eHealth Scenario, Quality Control round of Document
0.9.1	2021-07-13	DFKI	Completed Smart Manufacturing Scenario
1.0	2021-07-15	FTDS	Completed Smart Agriculture Scenario, 1st Version
1.1	2021-07-16	FTDS	Appendix Finalization, Abbreviations
1.2	2021-07-26	BYTE, UPM	Peer Review
2.0	2021-07-26	FTDS	Version for Quality Assurance
2.1	2021-07-28	INNOV	Quality Assurance
3.0	2021-07-31	FTDS	Final version V1
4.0	2022-09-29	FTDS	Updated ToC for V2
4.1	2022-10-20	DFKI	Updated KPIs for V2
4.2	2022-10-20	iSPRINT	Updated KPIs and Use Cases for V2
4.3	2022-10-20	CYBE	Updated KPIs and Use Cases for V2
4.4	2022-10-28	FTDS	Merging Pilot input, Update methodology and notation on changes for V2
4.5	2022-11-10	UPM	Peer review UPM – Ainhoa Azqueta
4.6	2022-11-10	BYTE	Peer review BYTE – Yannis Poulakis
4.7	2022-11-22	INNOV	Quality Check INNOV – Ariana Polyviou
5.0	2022-11-25	FTDS	Work Package Leader Reviewed – Niklas Franke

LIST OF ABBREVIATIONS

Abbreviation	Explanation
AI	Artificial intelligence
API	Application programming interface
BPMN	Business process management notation
CO2	Carbon dioxide
CSP	Cloud service provider
DoA	Description of action
F	Function
FaaS	Function as a Service
ha	Hectare
HPC	High performance computing
HW	Hardware
IoT	Internet of Things
KPI	Key performance indicator
Mb	Megabyte
ML	Machine learning
QC	Quality control
QoS	Quality of service
RT	Real time
RWD	Real-world data
SaaS	Software as a service
SW	Software
UC	Use case
V	Version

NOTE ON CHANGES COMPARED TO V1

This is the revised version of the Deliverable D6.3 and includes adjustments to the content compared to the first version based on experience gained during the adaptation of the pilots, as well as additions that have been newly developed. The following major changes have been made from Deliverable 6.3:

- Updated purpose of this Deliverable section in Executive Summary
- Addition of KPI Design Theory in Chapter 1.3
- Updates on the Methodology Diagram in the introduction of Chapter 2
- Addition of a text suiting the work stream of the Methodology Diagram in Chapter 2
- Updates on the Gantt Chart Diagram in the introduction of Chapter 2
- Update of Text in Chapter 2.1.6 due to former reference in D6.3 to D6.4.
- Change the Title of Use Case Scoping Workshop Description to "Use Case Scoping Methodology and Workshop Description" in Chapter 2.2
- Addition of the KPI Design Methodology and Workshop Description in Chapter 2.3
- Updated the Use Case chapters of the pilots with new findings
    - Smart Manufacturing Use Case #1.1 renamed from "Auto deployment of substitute services in the cloud" to "Deployment of substitute services in the cloud" in chapter 3.1.73.1.7
    - eHealth Use Case #2.1 renamed from "Auto Deployment of prediction Models" to "Deployment of Service" in chapter 3.2.7
    - eHealth Use Case #2.2 "Easier Management of the dynamic needs of the platform" in chapter 3.2.8 replaced by three Use Cases:
        - Use Case #2.2 "Model inference" in chapter 3.2.8
        - Use Case #2.3 "Patient phenotyping" in chapter 3.2.9
        - Use Case #2.4 "Data synthesis" in chapter 3.2.10
    - Add a fourth Use Case Smart Agriculture Use Case #3.4 "Scaling up" in chapter 3.3.9
- Replaced the Chapters of "currently used and desired performance measures" in chapters 3.1.9, 3.2.11 and 3.3.11 (Chapter numbers in eHealth and Smart Agriculture Pilots differ from chapter numbers in Deliverable D6.3 due to additions)
- Updates on the next Steps in chapter 4. Refer to next steps during Adaptation and Experimentation in the Pilots Journey
- Separation of the Appendix into the sections Use Case Scoping and KPI Design
- Addition of Figure 42 in the Appendix "Detail screenshot of the goal prioritization of the eHealth Pilot"
- Addition of Figure 43 in the Appendix "Detail screenshot of the quality check of initial formulated goals and the development into measurable performance results."

EXECUTIVE SUMMARY

The scope of this deliverable is to describe the use cases of the three pilots **Smart Manufacturing**, **Smart Agriculture**, and **eHealth**.
These use-cases cover three major areas of European everyday life and economic activity where the PHYSICS approach aims to improve agility and adaption by applying more advanced computing models and cover a wide and diverse range of available edge resources (e.g., small IoT sensors, mobile devices to powerful Edge nodes).

The purpose of *Task 6.2 - Use Cases Scenarios* is to

- Describe the use cases of the three pilots
- Identify the Use Case (UC) architecture
- Describe the relevant KPIs to measure the success of architectural progress in the rest of th e project.

To achieve these goals, an overall methodology was developed consisting of three streams: 1. KPI Design, 2. Use-Case Writing, 3. Architecture Description. These three streams were carried out in two phases: (a) Questionnaire, (b) in a series of workshops for all use-cases.

The **Smart Manufacturing** use case "Increased resilience and interplay" aims is to improve resiliency in the manufacturing domain by using Function-as-a-Service (FaaS) technologies and concepts provided by the PHYSICS platform.
The main objectives are described by:

(i) Auto-deploy substituted service in the cloud
(ii) High confidence quality control

The **eHealth** use-case "Personalized Monitoring and Collective Analysis" aim is to improve the performance and maintainability of the Healthentia platform, by using FaaS platform from PHYSICS.
The main objectives are described by:

(i) Optimize the systems ease of management, performance and scalability.
(ii) Optimize the deployment process of new updates to the system.

The **Smart Agriculture** use-case "Smart Precision Agriculture" wants to utilize the PHYSICS FaaS platform to enhance its greenhouse management scenarios.
The main objectives are described by:

(i) Improve the deployment process of simulation updates.
(ii) Ease deployment process of the system in a new greenhouse.
(iii) Ease the calibration of new plants measurements.

This deliverable "*D6.4 - Application Scenarios Definition V2*"

- Describes the application scenarios of the three different pilots
- Draws the borders of the Pilot systems, presents the different actors that are going to use the system in the pilots
- Expresses their individual goals against the system in their daily business, documents what actors benefit the most from FaaS and shows the existing architectures.
- Prioritize the TO-BE state of the use-cases
- Express the use-case scenarios in a unified structure
- Support the use-case scenario definition with a process map
- Designs and reports Key performance indicators used to evaluate pilot improvements on a use case level

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

1 INTRODUCTION

1.1 Context of the Deliverable

In the second revised version, this report documents the methodology and results of the selection of relevant application scenarios for each of the three pilot partners and documents the procedure and the results of a methodology for selecting suitable pilot specific KPIs. The presentation of the application-based methodology is taken from Deliverable D6.3 and adapted to the current status as adaptation and experimentation progress in the project. The methodology for developing KPIs, on the other hand, is a new addition to D6.3. Together with D2.5 and D6.2, this deliverable D6.4 provides the input for the second version of the application prototyping D6.6 as stated in Figure 1.

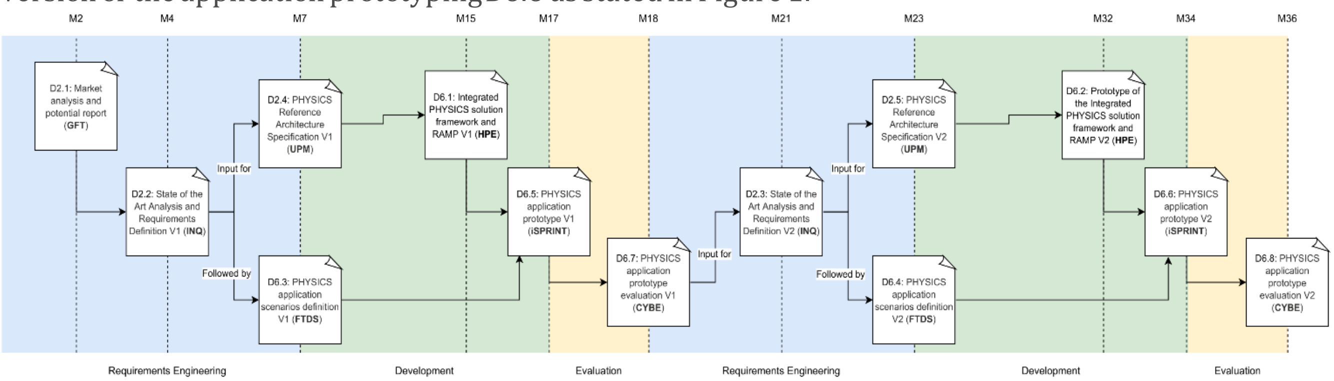

Figure 1 - Overview on how WP6 and WP2 Deliverables relate to each other in terms of the requirement engineering of the PHYSICS project

1.2 Use Case Theory

The use case comprises the set of ways to use a system to achieve a particular goal for a particular user [1]. The use case is the statement of the goal the primary actor has against the system and the system's behavior to deliver the goal[2]. The set of all use cases of a system yields all the useful ways to use the system and illustrates the value it will provide[1]. However, it only covers the behavioral aspects of a system[2]. The functional requirements of the system can ultimately be captured by describing how the system should respond to all possible requests as a set of use cases[3]. Whenever talking about requirements of a system, there are always one or more people or things (e.g. a service, a system or a physical component) interested in the behavior of the system. These are the stakeholders of the system[3]. The use case scenario builds the heart of a use case and focuses on how the system will be used to achieve the specific goal for a specific user also referred to as primary actor[2]. The result of the use case can either be a success or a failure[2]. One use case scenario is a sequence of interactions that happen between an actor and the system with the primary actor's goal as the intended result[2]. The interaction starts as a response to a trigger action and continues until the goal is achieved or abandoned[2]. Roles must be clearly articulated in it so that it is understandable who or what is doing something, the system itself included[3]. The sequence of steps that describes a use case scenario is explained in terms of the simplest way to achieve the goal. Exceptions that require deviating or alternate routes from the planned scenario are documented as an extension of the use case.[3]. The one and only primary actor is the stakeholder that interacts with the system to achieve a goal[2]. The actor is more of a role than a person, job description, or thing. It is also possible that an actor acts on behalf of a primary actor, or that a use case is automated and triggered by time, for example. If an external actor provides a service to the system, it is to be considered as a supporting actor or a secondary actor in the corresponding use case[2].

Careful consideration should be given to what is within and what is outside the scope of the use case[3]. If something is essential to the use case but is out of scope, it should be covered by another actor or use case and modeled as such[3]. It is also necessary to consider preconditions, as well as post conditions, which describe what conditions must be met before the scenario can begin and what conditions must b e met for the scenario to end with value. Success guarantees, like preconditions, are formulated as propositions that clearly describe what a state of the system would be that would satisfy one of the stakeholders' interests[3].

[1] Ivar Jacobson, Ian Spence, Kurt Bittner, Use-Case 2.0, Ivar Jacobson International 2011, pages 3-14
[2] Alistar Cockburn - Writing effective use cases, Addison-Wesley 1999, pages 2-204
[3] Jan Kettenis, Getting started with use case, Oracle white paper 2007, pages 2-14

1.3 KPI Design Theory

Companies have the reason to either maximize their operation, optimize activities or to make better decisions. Therefore, they use metrics which are per definition measures for the evaluation of efficiency, performance, advantage, or quality. Metrics are simply numbers. Per definition a metric has always one single dimension. If a metric would unite multiple dimensions, it would not be comprehensible why the metrics value is low. A measure should be tight to only one specific goal. To shape a broader pictur e of an operation it makes sense to have a set of metrics that measure all relevant facets of an entire operation. Key Performance indicators metrics that are used to gauge how good or bad a company or a business achieves its goals. Companies use a huge set of KPIs and rarely get rid of unnecessary ones. KPIs can have motivational effects, can convince others, and allow management from remote.

When wrong things are measured, it will lead to bad decisions, bad behaviour, and unhappy customers. People and specifically managers can also misuse metrics to their own benefit, to the disadvantage of others or to deceive others. It is crucial that users recognize that KPIs can provide the value to guide people for better decisions or to change behaviour and can be used to assess, evaluate, and analyse.

2 METHODOLOGY

The overall methodology consists of three work streams as depicted in **Error! Reference source not found.**, one for constructing KPIs, one for defining the use case scenarios and one for describing the AS-IS architecture. This method aims to define the requirements of the pilots for the upcoming PHYSICS architecture. The input of the documentation was collected and created in two iterative steps. The first step of the approach was a **questionnaire** that queried information that the pilot partners could provide from their existing pilot applications. The second part of the methodology was compiled in **workshops** because it concerned the future desired state of the pilot and required a more intensive discussion between the technical partners and pilot partners. The Use Case Modeling workshops were followed by the KPI Design workshops when the scenarios were defined.

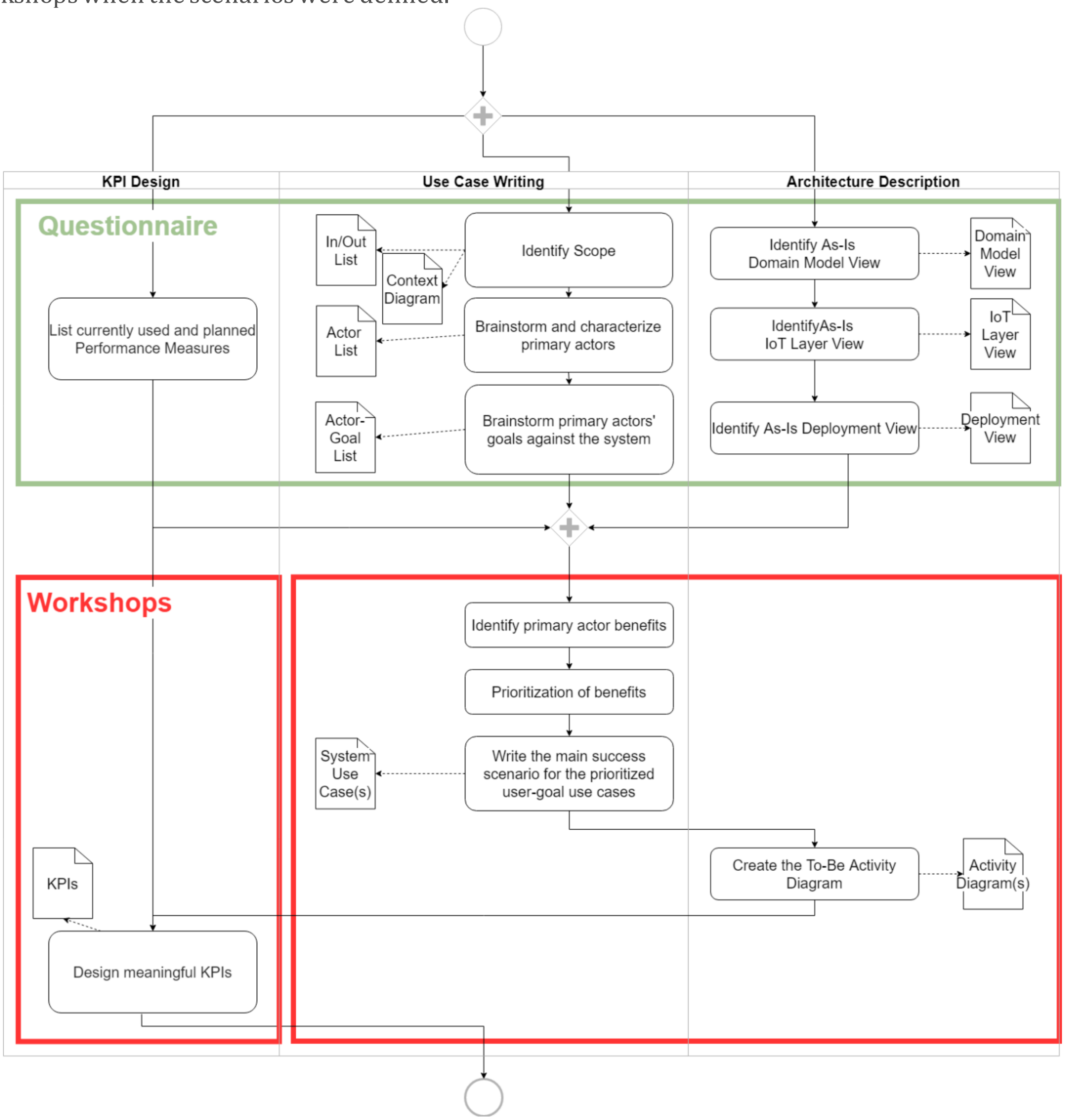

Figure 2 - Process View of the use case writing and KPI Design approach in Task 6.2

The first work stream "**KPI Design**" has the goal to design suitable key performance indicators for all three pilots. During this second version of the Application Scenario Deliverable, it was the main exercise to build up on the collection of the currently used performance measures of all three pilots from the first version of the Deliverable. This version contains individually designed key performance indicators per pilot.

The second work stream "**Use Case Writing**" targets writing use case scenarios in text form, for which several steps were necessary. The method started with drawing the borders of the system and showing the context of the pilot, followed by collecting all primary actors that are directly interacting with the system and characterizing them and brainstorming the actual goals that the actors have. Subsequently, the benefits that the different primary actors could have by a Function as a Service approach were collected during the workshops and a prioritization based on the benefits was made after that. Finally, the team described during the workshop the most beneficial use case scenario.

The third work stream "**Architecture Description**" collected the as-is architecture in three possible but not mandatory viewpoints within the Questionnaire and modeled activity diagrams of the architecture within the workshops for the individual use cases. The aim of the activity diagrams is to provide a good orientation for both technical and non-technical readers.

The following Gantt charts (Figure 3 and Figure 4) show the chronological sequence of the development of the content:

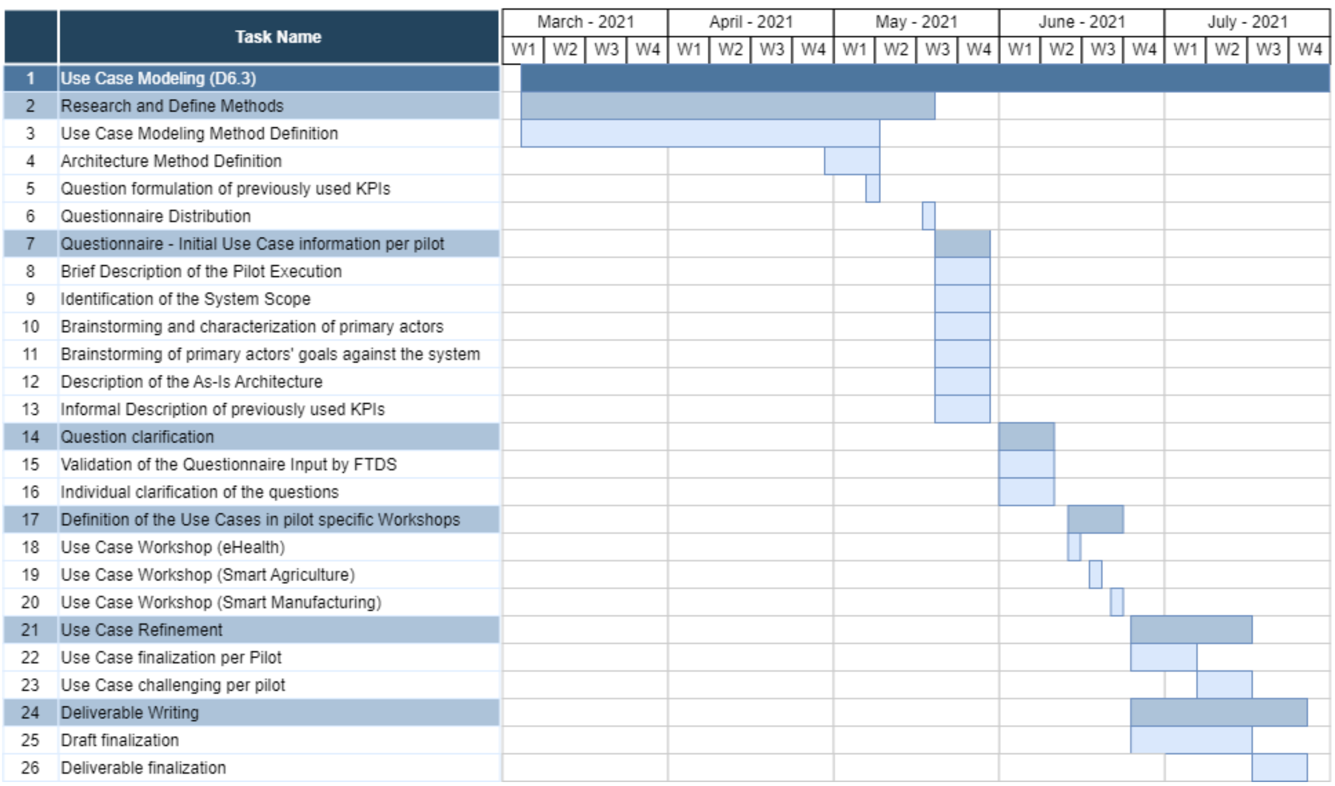

Figure 3 - Gantt chart of the work during the first version of this Deliverable

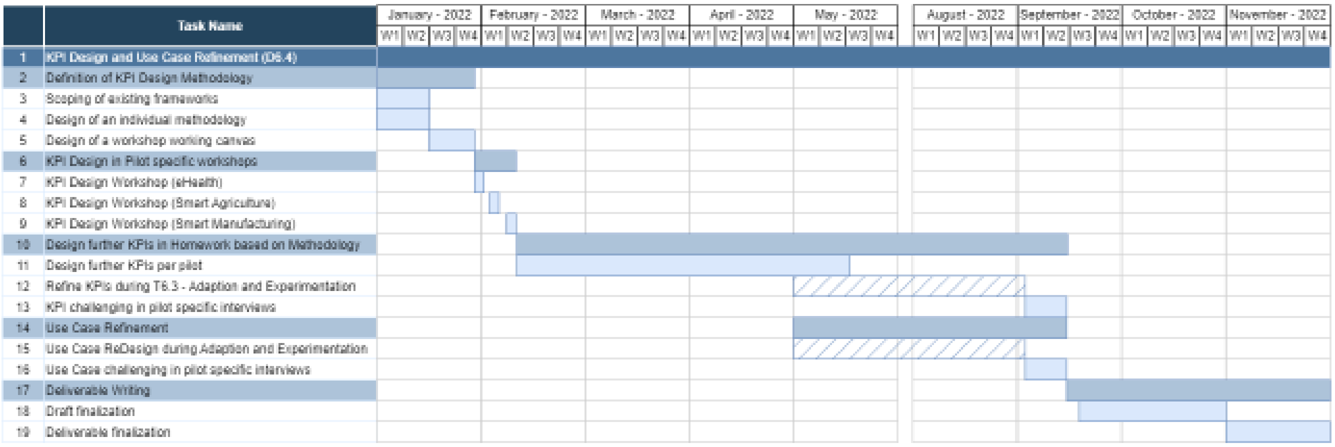

Figure 4 - Gantt chart of the work during the second version of this Deliverable

2.1 Questionnaire Description

Due to the high complexity of the pilot solutions in the PHYSICS project, as well as the specific domain know-how of the pilot partners, it is necessary to gather initial information on the pilots before conducting the workshop. The aim of this step was to create a common understanding of the pilots among the PHYSICS partners and to present the relevant pilot information in a generic, easily readable format. This step was also meant to help reduce the time needed for developing the use cases during the workshop. In order to extract and document as much relevant information as possible, a series of inputs in form of lists and drawings was requested from the pilot partners in the form of a questionnaire. The requested information included the aspects described in Section 2. The detailed answers to each requested questionnaire per pilot are presented in Section 3. Based on the inputs provided and documented, FTDS and the pilot partners defined the system use cases in a collaborative Workshop afterwards. In order to efficiently scope, define and document the pilot use cases, the following sequential procedure was utilized:

1. Supplying the initial Use Case information by pilot project partners
    a. Brief Description of the Pilot Execution.
    b. Identification of the System Scope (In/Out List, Context Diagram).
    c. Brainstorming and characterization of primary actors (actor list).
    d. Brainstorming of primary actors' goals against the system (Actors-Goal List).
    e. Description of the As-Is Architecture (Domain Model view is minimum needed).
    f. Informal Description of previously used Key Performance Measures and points for improvement.
2. Interpretation and clarification of open points by the Work Package Leader (FTDS) and Pilot Partners
    a. Validation of the Questionnaire Input by FTDS
    b. Individual clarification of the questions

In the following subsections we provide an overview of the questions that were posed to the use case partners and the scope of each.

2.1.1 Short Pilot Description

The descriptions in the sections 3.1.1, 3.2.1 and 3.3.1 aim to familiarize the readers of the Deliverable with the main features of the pilot, without going into specific details which will be covered further on. The description delivers a quick introduction of the goal of the pilot project, the main beneficiaries to the pilot, how the pilot is being currently used and what the expected behavior of the system with the utilization of PHYSICS components shall be.

2.1.2 The Design Scope

The Design Scope chapter serves to list all the relevant topics that are selected to be delivered and not delivered within the Pilot. The pilot partners provide a list of all the topics that came to their mind when thinking about the pilot and consider whether they are inside or outside the design scope of the pilot. An in/out list is used for topics related to both the functionality and the design scope of the system under discussion. The left column of the table (see **Error! Reference source not found.**) can include issues that came up when discussing the scope. The second column characterizes the item as software, hardware or a function. Furthermore, the pilots provide a context diagram. This step foresees to carefully model the boundaries of the system and all system actors. The focus of this context diagram is to pay attention on external factors and events to be considered in developing a set of system requirements. The diagrams show the existing system as a whole and inputs and outputs from and to external factors. An actor itself can be a human or a non-human. The diagrams will not provide any detail of the interior structure of the existing system under discussion. The context diagram shows the birds' eye view of the system under discussion as a black box within the ecosystem it is to be placed in. All system behavior and structural elements are completely inside the System under discussion and not covered by an actor.[4]

Table 1 - Design Scope Template

Item	Category	In Scope	Out of Scope
Example Item	SW	X	

2.1.3 The Actor List

The method conducted for the use case modeling followed the guidelines of writing in iterations. As use cases consist of actors and scenarios in their heart, the first actual exercise is to use a list to identify all possible primary actors[5]. This step is relatively important in order to characterize the actors and express the role in the context of the pilot[6]. The list that is used therefore tells the role name, the actor type and a short description what the actor needs to accomplish from the system functionalities. The actor type is also collected because it could be that secondary actors turn out to be primary actors at a later point. Primary actors are all those who interact with the system because they have an interest in achieving a goal. An actor can literally be anything having a behavior, and which capable of executing an IF-statement (e.g., a person, company, organization, computer program, computer system, hardware or software or both). If an external actor provides a service to the system, it should be considered as a supporting actor, or stakeholder in the corresponding use case, but not as a primary actor. With the creation of an actor list, the likelihood of satisfying the needs of the system users increases.

2.1.4 The Actor-Goal List

By definition, a use case is the statement of the goal that a primary actor has against the system in question[7]. That is why this exercise is a very crucial one in the use case modeling process. This exercise foresees to brainstorm all possible goals of the previously found primary actors towards the system. Here two aspects are important. One is to only consider the primary actors here. The other is to get the right level and formulate the goals, on the "User Goal" level, to express what the primary actor wants to complete. It is possible for an actor to have multiple goals against the system. In this case, a single goal of an actor cannot be a combination of two goals and therefore should not contain "and". The levels of the listed “Actor-Goals” must be what the main actor really wants from the system. All listed actor goals provide the basis for individual potential use cases.

2.1.5 Architecture Views

As specified in the overall methodology in chapter 2, the third working stream is intended to show the view of the existing and in-use Pilot System architecture of the pilots. The architectures are displayed in several suitable views.

[4] OMG System Modeling Language Version 1.6, 2019, pages. 241-242, Online in the web: https://www.omg.org/spec/SysML/1.6/
[5] Jan Kettenis, Getting started with use case, Oracle white paper 2007, pages 14-15
[6] Alistar Cockburn - Writing effective use cases, Addison-Wesley 1999, pages 32-33
[7] Alistar Cockburn - Writing effective use cases, Addison-Wesley 1999, pages 42

The **domain model** view is a web of interconnected objects. It inserts a layer of objects that models the business area in which the pilot is located. A simple domain model typically uses the vocabulary of the domain and incorporates the behavior and the data that are used to solve problems in that specific domain[8].

The **deployment view** models the physical deployment of artifacts on nodes[9]. The nodes appear as boxes, and the artifacts allocated to each node appear as rectangles within the boxes. One type of a node is a device node which is a physical computing resource like a mobile phone for example. Another type is an execution environment node which is a software computing resource that runs within a (computing) node that provides a service to execute software elements.

The pilots require for their real-time applications and services a high performing architecture. In order to show the node-to-node communication the architectures are put into the third architecture view which is an **IoT layer structure**[10].

2.1.6 Currently Used and Planned Performance Measures

The aim of this step is to initiate the definition of quantitative measures suitable for the comparison of the system using the FaaS approach and the current architecture as well as to document the process and results of the previous steps. This deliverable covers only the first step towards the design on new KPIs and collects only the ideas of the pilot partners in an informal way. These inputs are used later section 2.3 to design performance measures that are meaningful for the individual pilots.

2.2 Use Case Scoping Methodology and Workshop Description

After the questionnaires were completed, individual workshops were conducted for each pilot. The workshops were each attended by experts from the pilot partners, representatives of the technical work packages and FTDS as moderator. The workshops were scheduled as four-hour workshops and the agenda looked as follows:

Table 2 - Pilot Workshop Agenda

Timeslot	Topic
9:00 - 9:20	Welcome, round of introductions and brief look at the agenda.
9:20 - 09:45	Visioneering the personal success in three years.
9:45 – 10:00	Free Stage to the pilot
10:00 - 10:10	Short break
10:10 – 10:20	Introduction to FaaS from technical partner
10:20 - 10:40	Status Quo of Use Cases - consideration and prioritization according to other points of view
10:40 - 10:50	Short break
10:50 - 11:15	Explanation and prioritization of benefits
11:15 - 12:30	Use Case Definition in 2-3 groups in parallel
12:30 - 12:45	Use Case pitch and adoption

The questionnaire collected the actual task level goals, but most of the goals among them were not suitable for FaaS. Therefore, a 10-minute pitch about Function as a Service was given in order to inspire the group on the use of FaaS and create a common understanding of the possibilities of FaaS. After that, the team brainstormed how the primary actors could benefit from FaaS in their daily business. Then, the team voted what benefits they rate the highest among the benefits found. After the voting, the primary actor with the

[8] Fowler, Martin. Patterns of Enterprise Application Architecture. Addison Wesley, 2003, p. 116.
[9] Unified Modeling Language, Superstructure V2.1.2, 2007, p. 202
[10] Gilchrist, Alasdair. Industry 4.0 : The Industrial Internet of Things, Apress L. P., 2016. p. 94

most relevant benefit was identified and selected for the use case scenario description. The target of the description was to describe the achievement of one of his task-level goals with the visionary FaaS architecture of PHYSICS. The description was conducted on a Miro[11] canvas in an activity diagram. After the workshop, there was a summary and homework assignment to the pilot partners who participated in the workshop. In the homework, outstanding information was added that could not be completed during the workshop. Suh information included for example extensions to the main success scenarios, a second use case that would need too much time with the entire group or the dynamic as-is process view

2.3 KPI Design Methodology and Workshop Description

Developing a good evaluation metric requires commitment from everyone involved. There is no catalog from which managers can choose the perfect KPI. Rather, it is necessary to develop good and effective metrics, which are also lived, individually for the individual purpose. This kind of metrics must be measurable, achievable, easy to understand, fraud-proof and strategically aligned [12]. Unsuitable KPIs lead to incorrect behavior or do not lead to the creation of a meaningful basis for decision-making and the users create workarounds in order to appear to meet the KPI.

In relation to this project, we evaluated existing methodologies to create a suitable approach for our needs in a high-technology environment that aligns performance measurements for both technical components and the pilot's business.

During the first stage of evolution, we collected improvement requests for use cases as well as for the architecture with proposed KPIs from the pilot partners using a questionnaire (see Chapter 2.1.6). The results were evaluated and formed the basis for the eight-step workshops that followed (see Figure 5).

The workshops were each conducted with one or two representatives of the pilot partners, a technical partner, a project coordinator, and a moderator.

In step one of the workshops, attention was paid to the users of the future KPI on the business and operational level. We developed specific questions to identify the link between the operational level working with the KPI and the business level that needs to make decisions based on performance. The OKR concept according to Doerr [2018] proposes setting goals for a limited time horizon and based on this, assigning three to five key results to each. The key results should be measurable and serve to achieve the qualitatively formulated goal [13].

Therefore, during the second step, goals that are relevant for both the business and the operational level and what purpose a KPI should serve were identified. The KPIs can be used for orientation, improvement, or motivation. According to Goldratt's Theory of Constraints [1990], a KPI should be assigned to the bottleneck of a process if possible [14]. Therefore, the most relevant goals were prioritized by a vote of the workshop members and represented the most urgent bottlenecks of the pilot partner.

In step three, the prioritized goals were reformulated into performance results, analogous to the OKR concept. Specific tests based on Barr's [2014] performance measurement process were used to transform the goals. The first test ensured the formulation of a goal as an achievable outcome. The second test ensured that non-words were replaced with clearly defined language and a third test ensured that the result contained only one focus. Passing these tests qualifies the target as a valid performance result. When multi-

[11] https://miro.com/

[12] Eddie Davila. 2018. Die Unternehmensleistung messen. Retrieved August 12, 2022, from https://www.linkedin.com/learning/die-unternehmensleistung-messen/warum-wir-messen?autoAdvance=true&autoSkip=false&autoplay=true&resume=false&u=83641554

[13] J.E. Doerr. 2018. Measure What Matters (1st. ed.). Penguin Random House, London, England

[14] Eliyahu M Goldratt. 1990. What is this thing called theory of constraints and how should it be implemented? (1st. ed.), North River Press, Croton-on-Hudson, N.Y., United States

focus is detected, the results are separated and then treated individually as separate performance results, as shown in the example in Figure 6.

The fourth step was also selected from Barr's [2014] performance measurement process, since achievement of a result cannot be evidently established in all cases, but only by observing an indicator used as sensory evidence [15]. If a performance result alone cannot provide sensory evidence, evidence must be found at this point. The key performance indicator is formulated in unambiguous language in writing from clearly obtainable evidence. The following steps characterize the handling of the newly developed KPI.

In the fifth step, it is shown how the KPIs are recorded and whether the reporting interval of the KPIs is in relation to the effort involved in creating the KPIs. It checks whether the parts of the KPI can be measured directly, calculated from other data or pulled from another existing indicator - or partly a combination of these possibilities.

Step six has two parts. The first, starting from the basis of the KPI components, looks at the mathematical relationships between the components to build the KPI. In the workshop, sticky notes on a virtual whiteboard were used to create the formula. The second part deals with the definition limits between a desired optimum, an acceptable working range and a non-tolerable range.

Since the KPI needs to be assigned a responsible owner, step seven checks whether an existing process can take responsibility for the KPI integration or whether a new one needs to be defined. If an existing process can share responsibility for the KPI integration, it is checked which adjustments need to be made in the process and who is responsible for this. The overarching goal of this step is to turn the KPIs into elements of lived processes.

Number eight is essential for the actual implementation and use in the company, because the collection of KPIs is always associated with people. There are those people who work with the KPI, others who are responsible for the process that measures the KPI. Others who report based on the KPIs and possibly those whose rewards are linked to the KPI. During the workshop, participants from the relevant areas of the company are made aware of how KPIs can be misused for their own benefit and to the detriment of others. This task protects against misuse of the KPIs by covering possible scenarios with a complementary KPI. For example, quantitative results can be supplemented by qualitative ones.

[15] Stacey Barr. 2014. Practical performance measurement using the PuMP Blueprint for Fast, Easy, and Engaging KPIs (1st. ed.). The PuMP Press, Samford, Queensland, Australia

The questions developed are assigned to the individual before mentioned steps below:

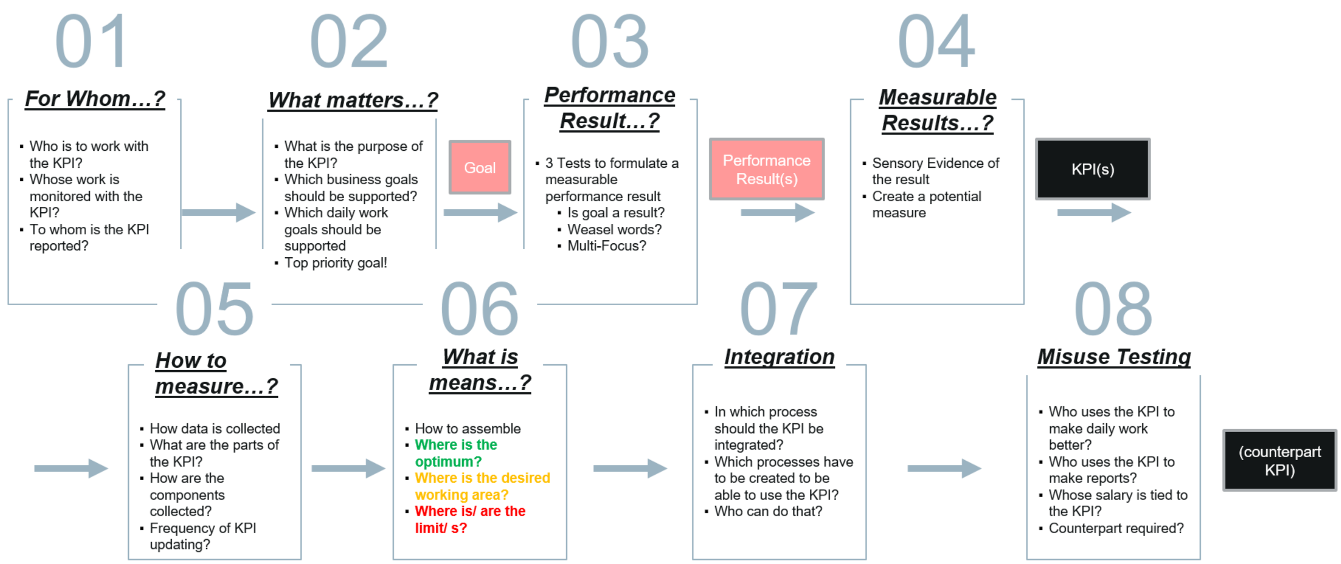

Figure 5 - KPI Design Workshop Steps

Figure 6 shows a screenshot of the whiteboard from the eHealth Pilot's KPI design workshop. The numbers refer to the eight steps of the KPI design method described above. The goal formulated first con tains two focus points and therefore had to be divided into two goals in step 3.) and formulated into two KPIs in step 4.). In step 8.), a complementary KPI had to be developed after the abuse test, which is marked by c.). Thus, this representation shows the development of three KPIs.

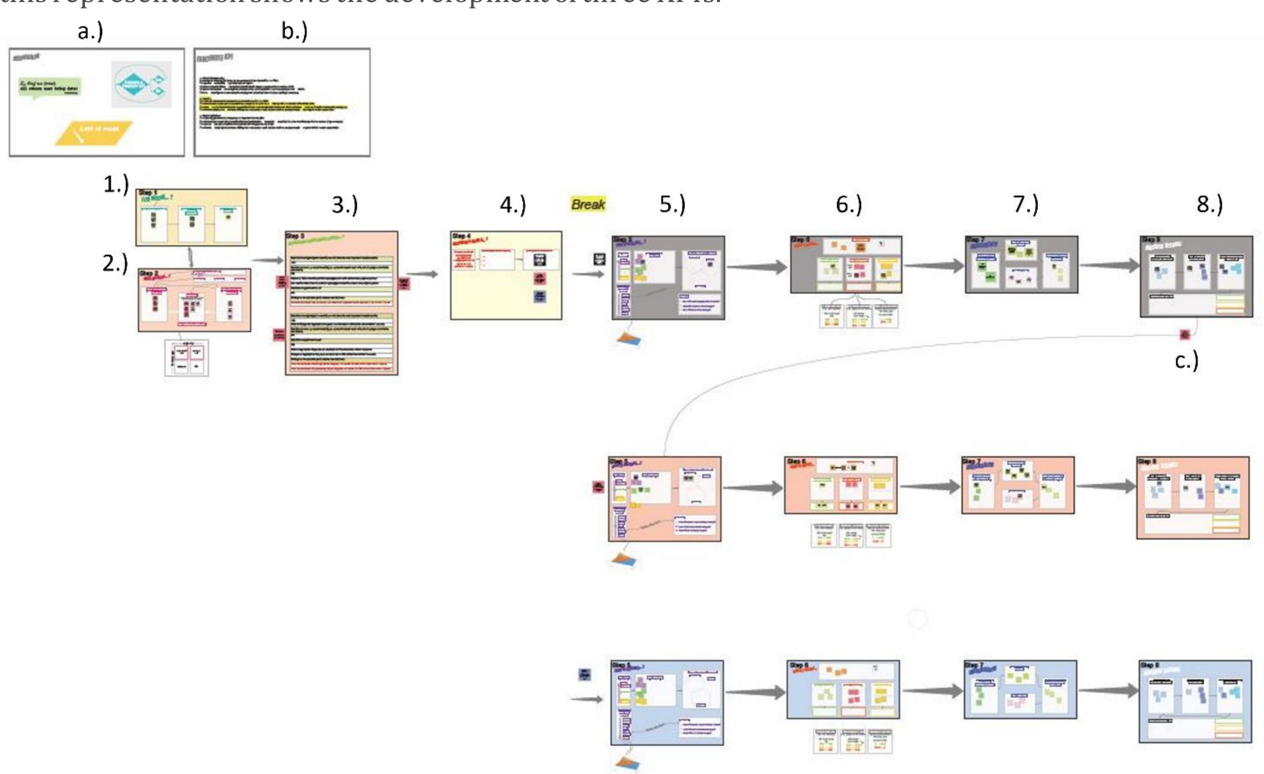

Figure 6 - A whiteboard from a KPI design workshop

.

3 PILOT APPLICATION SCENARIOS

3.1 "Smart Manufacturing" Pilot

3.1.1 Smart Manufacturing Pilot Description

SmartFactory-KL brings research and industrial companies together in order to implement demonstrators in accordance with Industry 4.0-compliant standards and to test new technologies. The demonstrators are manufacturer-independent to avoid vendor lock-in problems and highly flexible. The newest demonstrator (Production Level 4, Figure 7) is designed to be highly autonomous. With "lot size one" production, it assembles user-customized USB pen drives using different autonomous and interoperable modules, each dedicated for a single step of the production. Modules are independent and controlled by themselves in case it is their turn to continue the production. The software architecture of the overall infrastructure maintains how the product is produced, by generating a recipe for manufacturing, scheduling the products by their priorities, production time, etc. The software architecture is designed with a service-oriented approach, which enables decoupling and an easier conversion into FaaS approach in PHYSICS project. Each software service is registered in a registry as soon as it is deployed. Later, software services communicate with each other querying this registry for the endpoints of a specific service. This registry sends the metadata of the requested service, and the communication is then performed between requestor and the requestee.

Figure 7 - Production Level 4 Demonstrator which will be enhanced within the PHYSICS Project

The following three conditions are not currently supported in the current behavior of the system, which service-oriented architecture is shown in Figure 8:

- There is no redundancy in case of a software service failure occurs. Its functionality is lost, and this can affect the production.
- It is not possible to retrieve service metadata if the service registry fails.
- The AI, in particular the visual quality check, computations are always performed locally. If the AI computation takes longer, it cannot be offloaded to a more available server or to the Cloud.

With the PHYSICS project it is planned to cover these three aspects to enable load balancing as well as redundancy and increase reliability. The load balancing and service monitoring features must also be always available (99% uptime).

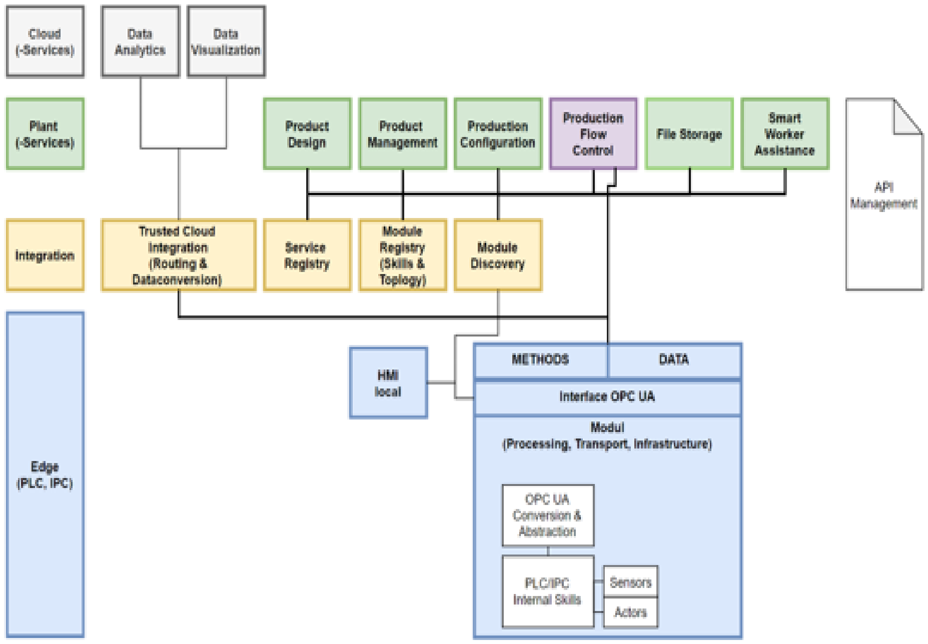

Figure 8 - The high-level software architecture of the Production Level 4 Demonstrator

3.1.2 Smart Manufacturing Design Scope

The Smart Manufacturing use cases will include several components which are Software (SW), Hardware (HW), or in general Functions (F). The following list specifies which of them will be used and/or improved and which of them are out of scope.

Table 3 - Scope In-/Out-List for the Smart Manufacturing Pilot in PHYSICS.

Item	Category	In Scope	Out of Scope
Product Configuration	SW	X	
Quality Assurance Module	HW	X	
Edge Server	HW	X	
Production Flow Control	SW		X
Deployment of AI functions	F	X	
Training of AI models	F		X
Product Design	SW		X
Product Management	SW	X	
Smart Worker Assistance	SW		X
Service Registry	SW	X	
Module Registry	SW	X	
Module Discovery	SW	X	
Adjustment of transport rail parameters	F		X
Software Deployment	F	X	
Quality Control	SW	X	
Data Refuelling	HW		X
Transport of the product via transport rail between stations	F		X
Filling raw materials	F		X
Any non-software parts of the system	HW		X
Maintenance of hardware	HW		X

Recognize the new known service deployments in production	F	X	
New software services	SW		X
Transfer of material/data from external systems/suppliers	F		X
Software service monitoring, load balancing, and orchestration	SW	X	
File Storage	SW	X	
Database (Redis)	SW	X	

Figure 9 below expresses the boundaries of the SmartFactory System and all the external actors. The diagram also shows the external factors and events to be considered in developing functional requirements. The diagrams show the existing system as a black box and inputs and outputs from and to external factors with arrows.

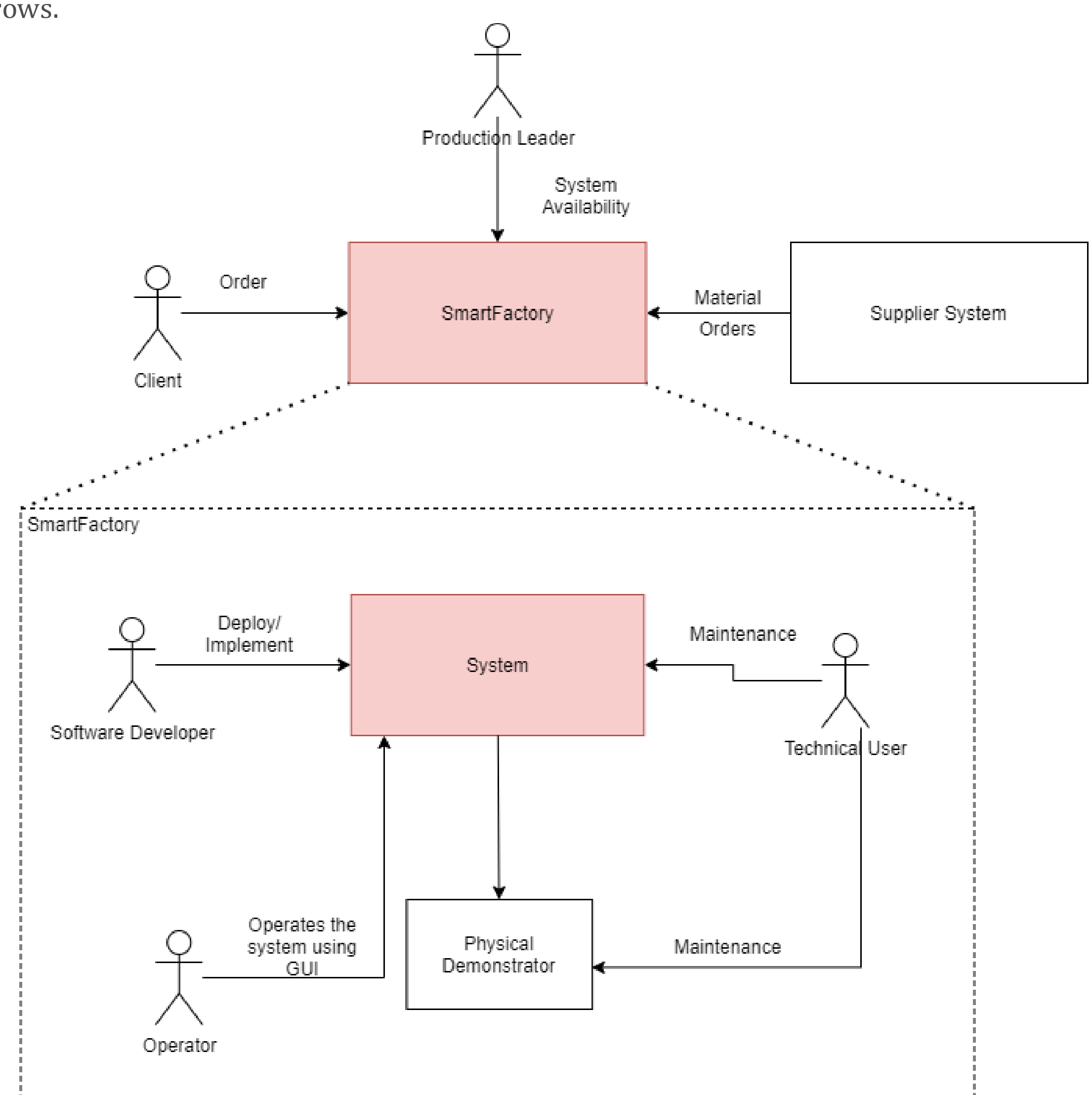

Figure 9 - User-defined Context Diagram for the Smart Manufacturing Pilot in PHYSICS.

3.1.3 Smart Manufacturing Actor List

Table 4 lists below all the different actors that are relevant for the use cases are enlisted. The natural persons are depicted in Figure 9 above.

Table 4 - Actor List describing the different Actors that are relevant in the Smart Manufacturing Pilot.

Name of the role	Role Type	Characterization of the skills and the job of that actor type
Technical User	Primary Actor	Person who fixes the system in cases of software or hardware problems.
Operational User	Secondary Actor	Worker that operates the machine using on-site interfaces. Informs the technical user in cases of emergency.
Software Developer	Primary Actor	Person who provides services related to production. No deep knowledge of FaaS.
Production Leader	Stakeholder	An organizational person who seeks a high uptime and high production rate.
Customer	Stakeholder	An external person who orders new products.
Physical Demonstrator	Secondary Actor	A manufacturer independent system, which completes various tasks in order, to complete a product. Sends status messages to the software which orchestrates the production.
Local Quality Control Service	Primary Actor	An AI-based quality control service that checks for product defects automatically.
Local Production Flow control	Primary Actor	A service that orchestrates production services ("skills") of all production modules in the required order.

3.1.4 Smart Manufacturing Actor-Goal list

Table 5 - Extended information on the use case actors is provided in the Actor-Goal List.

Primary Actor	Task-Level Goal	Brief Description
Maintenance User	Deployment of substitute services.	The user expects autonomic deployment of services in case the services fail to serve the production system, to keep the uptime high.
	Monitor service statuses and the reason for failure.	The user tracks the services and their statuses to get an overview of system load and react to it in case of a problem.
Software Developer	Deploy a software related to the production in the local VM.	The user implements their service so that it can automatically be integrated in the system and can benefit from FaaS.
	High confidence quality check	When confidence is lower than a threshold, a more complex quality check must be deployed in the PHYSICS cloud platform. Due to FaaS' pay-per-use it is always ready but, it does not cost until it is actually used.

3.1.5 Smart Manufacturing As-Is Architecture Views

This section presents all three architectural views that were presented in the introduction in Section 2.1.5 above. Figure 10 shows the current domain model view which models the business area in which the Smart Manufacturing pilot is located.

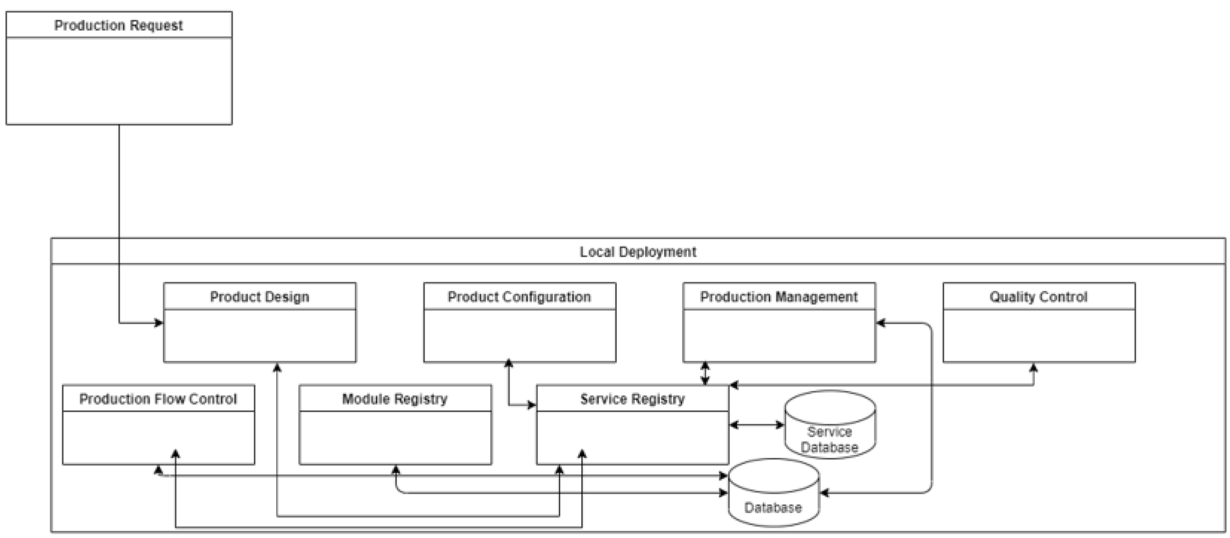

Figure 10 - Domain Model View for the Smart Manufacturing Pilot in PHYSICS.

Figure 11 below shows the IoT Layer View consisting of the Smart Manufacturing Pilot[16]. The Layers show the present components and protocols in the IoT structure.

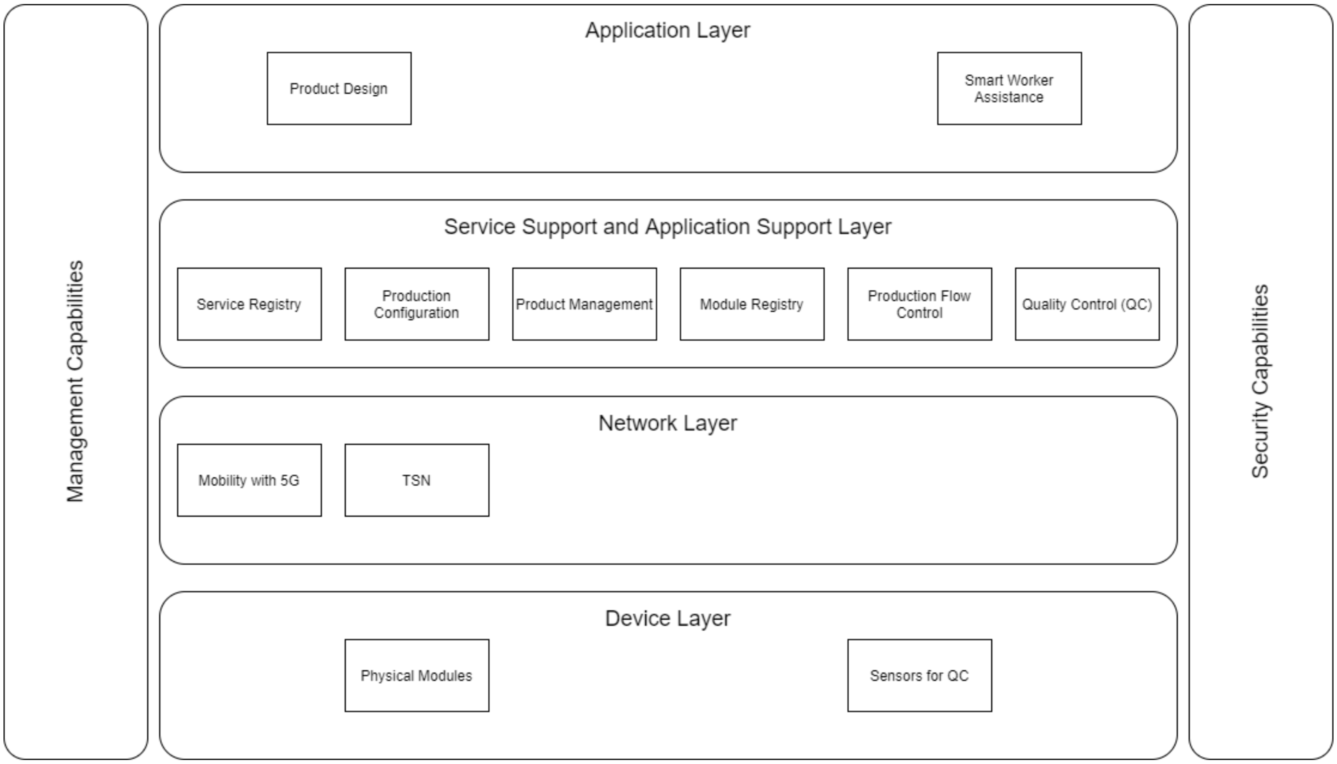

Figure 11 - IoT Layer View for the Smart Manufacturing Pilot in PHYSICS.

[16] Gilchrist, Alasdair. Industry 4.0: The Industrial Internet of Things, Apress L. P., 2016. p. 94

The currently used Deployment View of the Smart Manufacturing Pilot is depicted in Figure 12 below. It shows the physical deployment of artefacts on nodes. Nodes in this case are devices and execution environments.

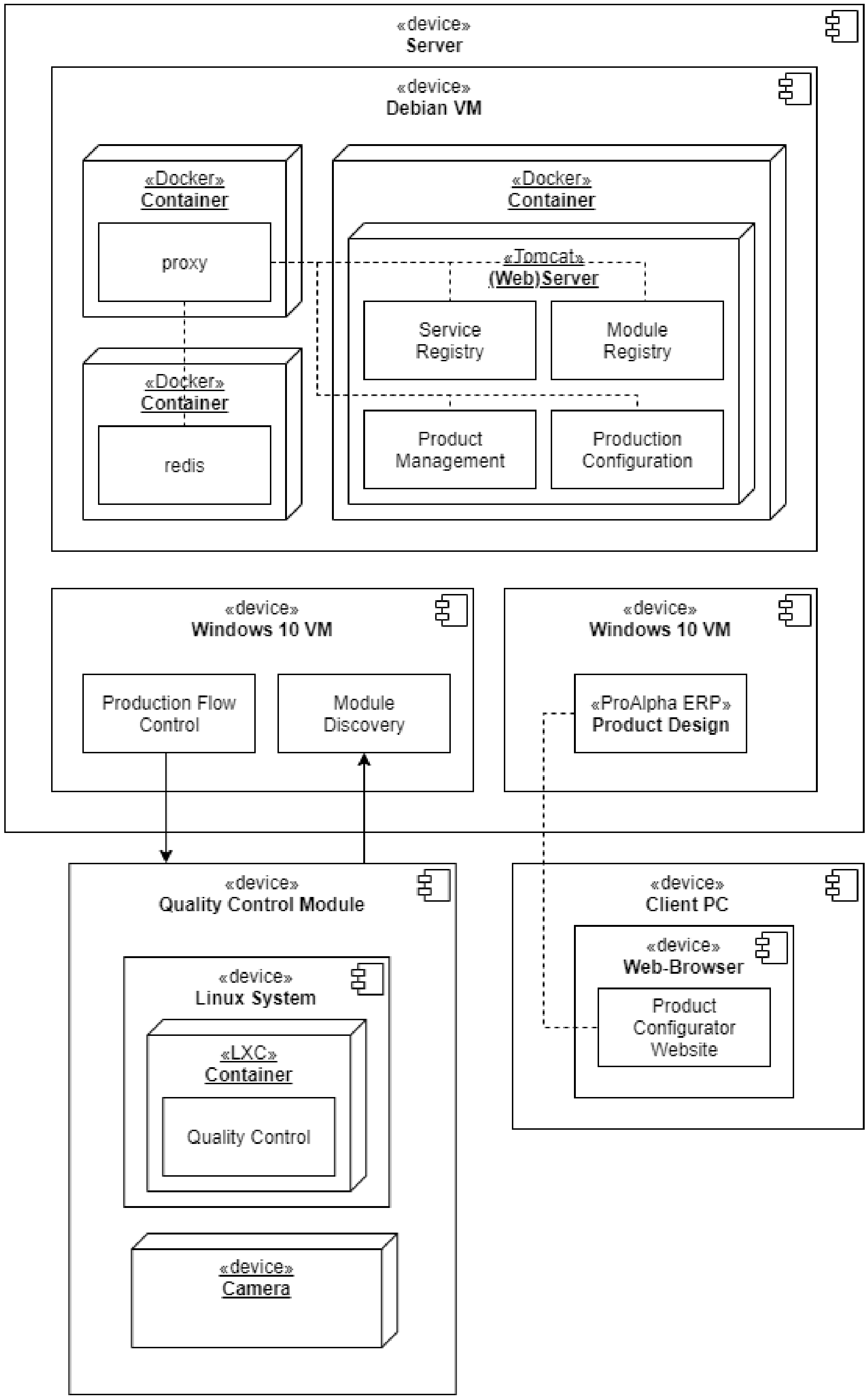

Figure 12 - Deployment View for the Smart Manufacturing Pilot in PHYSICS.

3.1.6 Smart Manufacturing Actor Benefits

With the use of FaaS patterns and the overall PHYSICS architecture there are some primary actors that are interacting with the SmartFactory software that would benefit fundamentally. In the workshop the team discussed what actors would benefit the most. Table 6 shows the identified benefits and the prioritization that was conducted in the workshop. The major benefit identified as *"More stable and reliable system"* was prioritized for the use case modeling in the following two sections. During the use case modeling was noticed that the maintenance user was actually not the primary actor because they do not interact with the system in this particular use case. Instead, two software components were identified as the primary actors (Local Production Flow control and Local Quality control service). The maintenance user was now seen as a stakeholder.

Table 6 - Prioritization of the benefits for the primary actors with FaaS in the Smart Manufacturing Pilot

Primary Actor	Benefit from FaaS	Election
Maintenance User	▪ Better control of the system ▪ More granular control	0 votes
	▪ More stable and reliable system ▪ Increase of reliability of the system	6 votes
	▪ Less maintenance (No manual restart of the services) ▪ Easier identification of problems	7 votes
Software Developer	Reduced expert knowledge on FaaS (less development time)	3 votes
	More granular responsibility	0 votes
	Improved Development: ▪ Faster testing, ▪ Easier adding of functions, ▪ Easier teamwork with other developers ▪ Easier replacement of underperforming components	8 votes

3.1.7 Smart Manufacturing Use Case #1.1: Deployment of substitute services in the Cloud

The first use case that was identified as relevant to implement is the use case “Deployment of substitute services”. The use case was chosen from the previously prioritized benefit “more stable and reliable system”. The main goal of the use case is to increase productivity by improving the reliability and decreasing unplanned downtime of the production. Figure 13 shows the activity diagram of the use case with the currently used architecture.

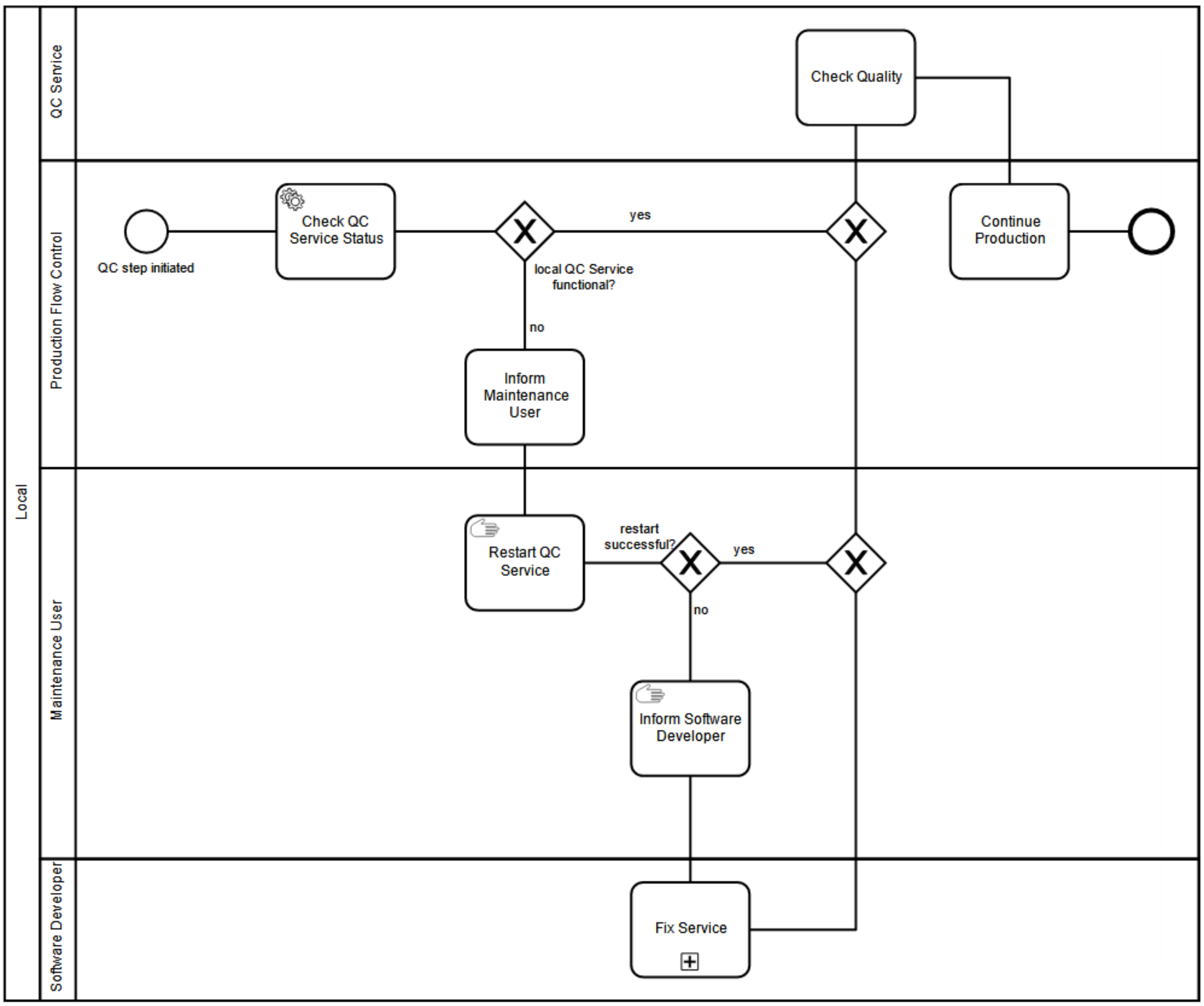

Figure 13 - Smart Manufacturing Use Case #1.1 AS-IS BPMN Diagram

As specified in the overall methodology (Figure 2), the target of the second stream is to describe the system use case in text form. The following table shows the first Smart Manufacturing use case.

Table 7 - Smart Manufacturing Use Case #1.1: Deployment of substitute services in the Cloud

Use Case #1.1	**Deployment of substitute services in the cloud**	
Brief Description	Local Quality Control Service fails e.g., due to a server error. A previously defined PHYSICS-enabled application is readjusted and deployed in the cloud-edge-continuum automatically. The system is always operational. Maintenance user comes to work and checks the systems health in order to view the service status. The maintenance user gets notified that the local service failed. He likes the system, because he can solve local problems in his own pace and is not rushed. He knows that his boss will not be unhappy, because the production is not negatively affected.	
Context of use	Local Quality Control Service fails (i.e., Server failure)	
Scope	SmartFactory software and PHYSICS platform.	
Level	Task Level	
Primary Actor	Local Production Flow control (local PHYSICS platform possible)	
Stakeholder & Interests	**Stakeholder**	**Interest**
	Business Owner	No Downtime
		Minimal Cost
		Reduced product returns
	Production Leader	Easier problem solving
		No unplanned Downtime
		Reliable system
	Customer	No product delays
		No defective product
	Maintenance User	Softer deadlines
	Software Developer	Less distractions
		Softer deadlines
Preconditions	▪ Software developer has selected the appropriate patterns to enable more robust and continuous deployment and operation. ▪ Software Developer has the “FaaSified” service deployed in the PHYSICS platform. ▪ Software Developer has specified the hardware capabilities to find a similar capability in the cloud (or a different edge device in the factory in the case that PHYSICS is deployed locally).	
Success End Scenario	Quality control service continues with the deployment in the cloud	
Failed End Protection	Re-adjustment is not possible to be performed	
Trigger	Notification from monitoring service that the local Quality Control service is not functional.	

Success Scenario	Step	Action	System Reaction
	1	The local Production Flow control calls the Quality Control function in the PHYSICS platform in the cloud.	▪ PHYSICS platform chooses one of the available solutions and adjusts the deployment. ▪ The service gets deployed on the new instance ▪ PHYSICS platform adjusts combined cloud-edge-deployment ▪ Computation in the PHYSICS platform and sending back the results to the local Production Flow Control
	2	Production Flow Control receives results from the PHYSICS Cloud platform and acknowledges the PHYSICS cloud platform about the reception of the results.	Cloud platform instance can be shut down.
Extensions	**Step**	**Branching Action**	
	1a	No internet connection	
	1b	Failure is not addressable (e.g. error in the code)	

Figure 14 below shows the activity diagram of the use case that the future architecture should support. The diagram is supporting the Use Case text of Table 7.

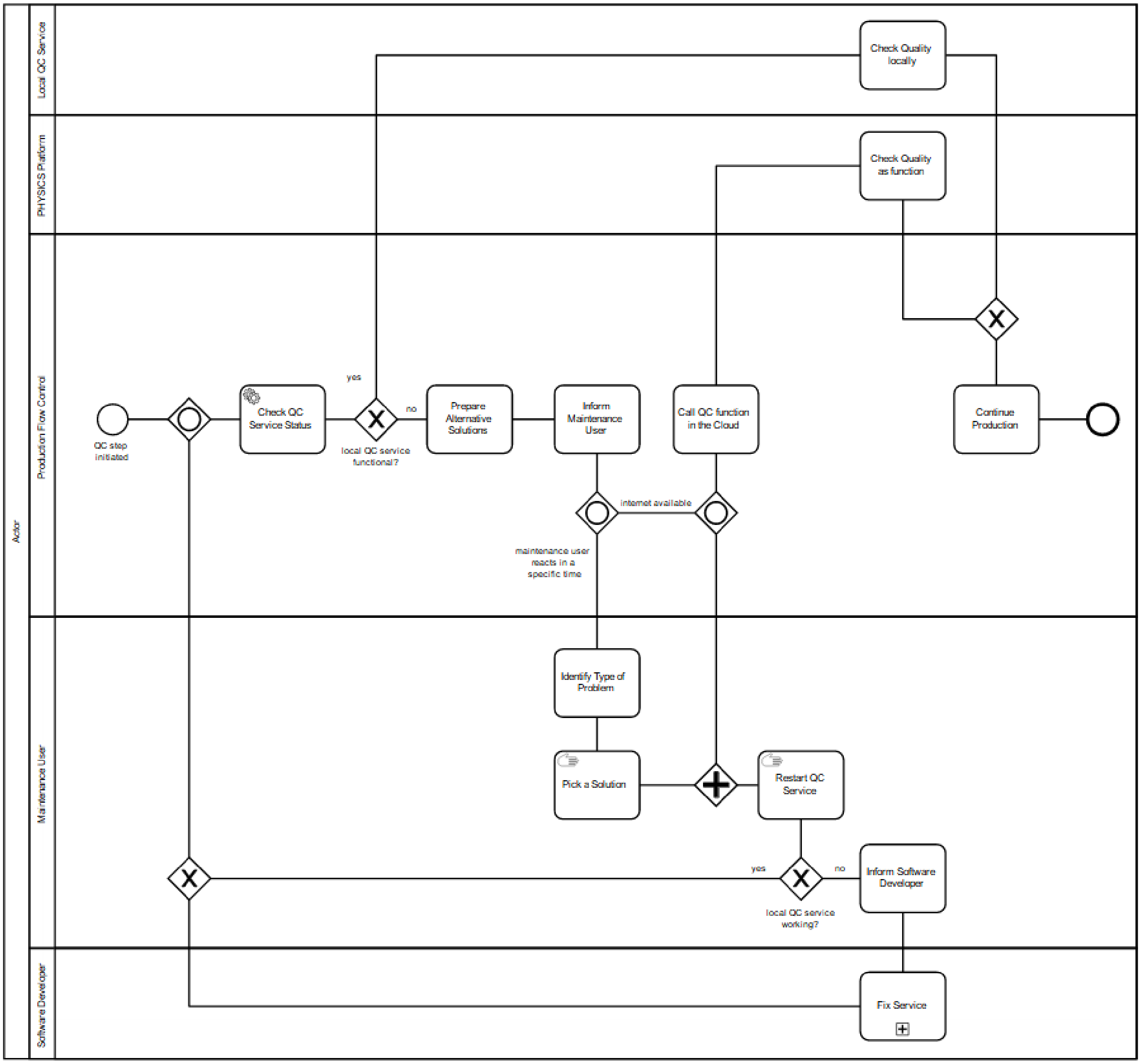

Figure 14 - Smart Manufacturing Use Case #1.1 TO-BE BPMN Diagram

3.1.8 Smart Manufacturing Use Case #1.2: High Confidence Quality Control

The second use case identified as relevant to implement is the Use Case "High Confidence Quality Control". The main goal of this use case is to have high quality products. Figure 15 shows the activity diagram of the use case with the currently used architecture. Table 8 shows the to-be scenario of the second Smart Manufacturing use case.

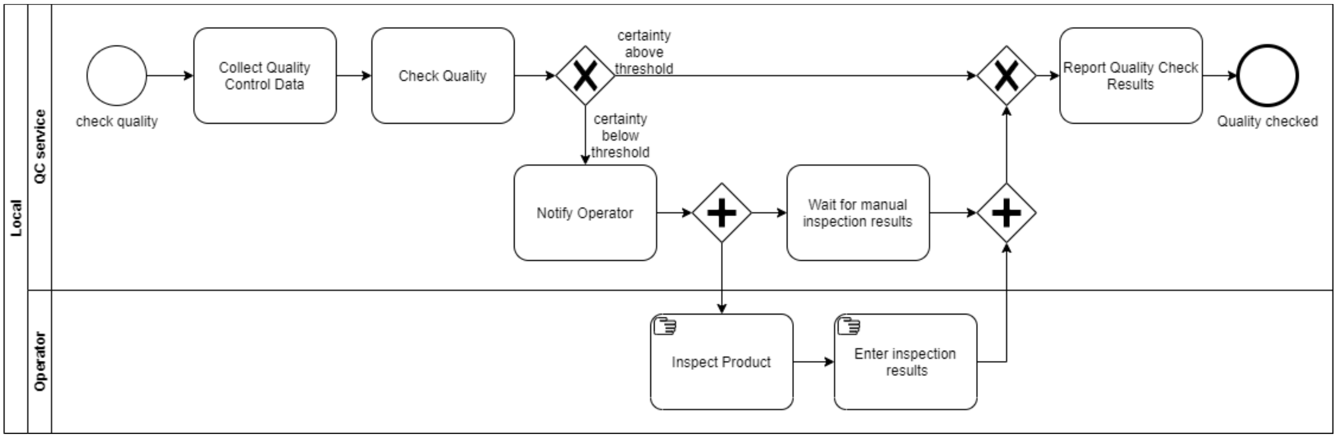

Figure 15 - Smart Manufacturing Use Case #1.2 AS-IS BPMN Diagram

Table 8 - Smart Manufacturing Use Case #1.2: High Confidence Quality Control

Use Case #1.2	**High Confidence Quality Control**	
Brief Description	The local quality control computation power is limited relative to the cloud. In case of a low confidence level of the local quality check the task of the defect inspection is shifted into the cloud because of a more complex check. The business owner will have less returns and potentially less rejects due to high QC accuracy and precision, but he will also benefit from the pay-per-use model of the FaaS service. The operator benefits because the new system has additional QC capabilities and hence less manual inspection is required.	
Context of use	Confidence level of quality check on edge is below the threshold and needs a more complex computation in the cloud.	
Scope	SmartFactory software and PHYSICS platform.	
Level	Task Level Use Case	
Primary Actor	Local Quality Control Service	
Stakeholder & Interests	**Stakeholder**	**Interest**
	Business Owner	Reduced product returns
	Customer	No product delays
		No defective product
	Operational User	Less manual quality inspections
Preconditions	▪ More complex quality check is available in the cloud ▪ Software developer has selected the appropriate patterns to enable more robust and continuous deployment and operation ▪ Software Developer has specified the hardware capabilities to find a higher capability in the cloud with possibly GPU acceleration.	
Success End Scenario	Have highly accurate and precise QC	
Failed End Protection	Reject product	
Trigger	▪ Confidence level of local Quality Control service is below threshold. ▪ Response time exceeds the defined time per task.	

Success Scenario	**Step**	**Action**	**System Reaction**
	1	Data required for QC is forwarded to Cloud	Computation in the PHYSICS platform and sending back the results to the local Quality Control service
	2	Check the threshold of the received results from the PHYSICS platform	Report Quality Check results
Extensions	**Step**	**Branching Action**	
	1a	Error: No internet connection would require a manual inspection	
	2a	Error: Low certainty result from FaaS platform would require a manual inspection	

Figure 16 shows the activity diagram of the use case "High Confidence Quality Control" that the future architecture should support. The diagram is supporting the use case text of Table 8.

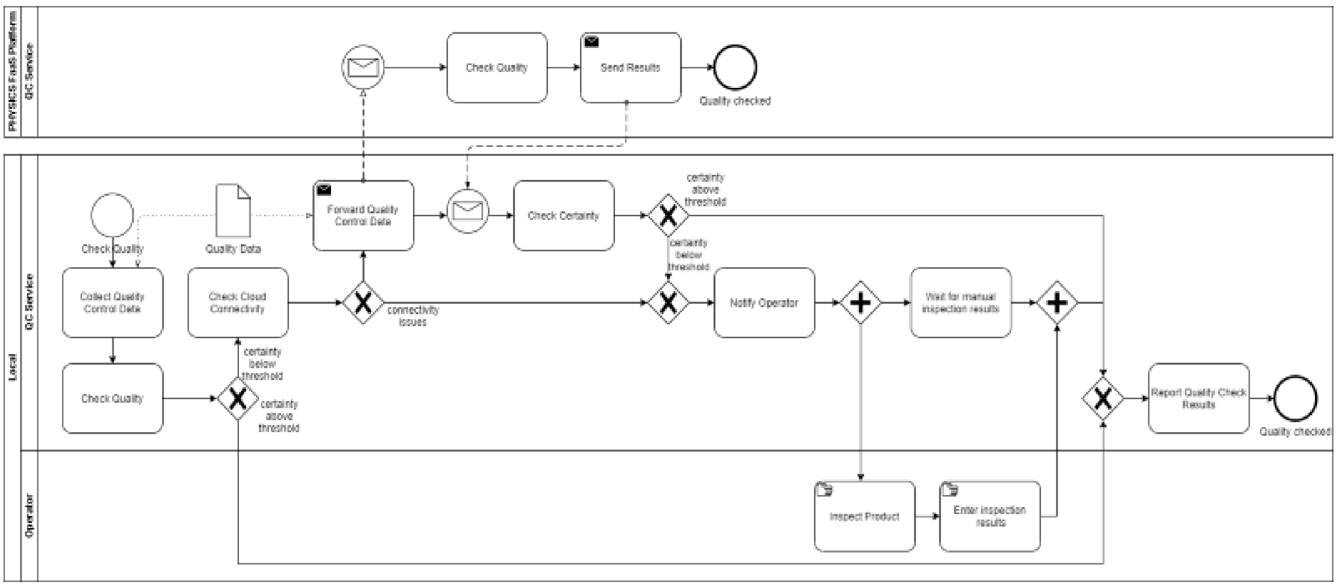

Figure 16 - Smart Manufacturing Use Case #1.2 To-Be BPMN Diagram

3.1.9 Smart Manufacturing Key Performance Indicators

The following key performance indicator were defined for the Smart Agriculture pilot based on the methodology presented in chapter 2.3. The Evaluation of the KPIs in the current infrastructure has been reported in Deliverable 6.7 "PHYSICS application prototypes evaluation V1"and will be updated later in the project in Deliverable 6.8 "PHYSICS application prototypes evaluation V2".

- **KPI01 - Time for handling a request:** Measures the average request/response time for 30 requests with PHYSICS and before PHYSICS.
- **KPI02 - Scalability:** Measures request/response time for up to 30 parallel requests.
- **KPI03 - Number of software logs per QA'ed product:** Counts the number of errors with and without failover functionality of PHYSICS while using QC function.
- **KPI04.1 - Reliability:** Measures the reliability of the QC function using MTBF (mean time between failures) (number of operational hours / number of failures)
- **KPI04.2 -** Availability measures the availability of the QC function over the period on a weekly base (number of operational hours/ total number of intended machine usage hours per week)
- **KPI05 - Interference latency:** Like KPI01, except it only calculates the time to complete the QC action, without considering the overhead of OpenWhisk.
- **KPI07 - Number of PHYSICS invocations in software logs per QA'ed product:** Similar to KPI03, but this measures the ratio between the invocations on PHYSICS and the local QC service.
- **KPI08 - Cost reduction:** The bill based on the time multiplied by the cost/time.

There were also some previously ideated KPIs dismissed for further measurement:

- KPI06 - **Data protection**: Indicates how the transferred data is protected
- KPI09 - **Performance benefits**: Although for the manufacturing Use Case, the reliability (KPI04) is more important than performance benefits, this KPI will inspire us to further use FaaS technology in our upcoming demonstrators as well as the functionalities of the existing demonstrators. (Clarification: KPI01-KPI08 are already performance indicators)

3.2 "eHealth" Pilot

3.2.1 eHealth Pilot Description

The aim of Pilot number two: "Personalized Monitoring and Collective Analysis" is to improve the performance and maintainability of the Healthentia platform, developed by Innovation Sprint, by using Function-as-a-Service (FaaS) technologies as provided by the PHYSICS Platform for some of the smart services on offer. Healthentia is an eClinical Software-as-a-Service (SaaS) platform, consisting of a mobile app for patients/citizens, a web portal for healthcare professionals and researchers, and a server-platform for data storage and processing. The Healthentia SaaS main offering is to allow clinical sponsors to define their own studies, invite participants to join those studies, monitor participants' progress and enable various smart services that provide data analysis or virtual coaching functionalities. Healthentia's main clients are sponsors that want to run clinical studies and use Healthentia to collect Real-World Data (RWD) to obtain a better picture of their clinical trial participants. RWD includes wearable data (e.g. physical activity, sleep), self-reported events or symptoms (e.g. pain, cough, fatigue), and answers to questionnaires that can be defined within the web portal. Innovation Sprint is in the process of extending the offering of Healthentia from clinical trial services to digital therapeutic services, integrating features that offer direct health- and lifestyle support to end-users (patients/citizens).

Besides the many organizations that define and run their own studies, Innovation Sprint offers the Healthentia mobile application freely to the general public. In essence, users who download the mobile app and are not invited to join a specific sponsor-driven study, can enter in the *default* study. The purpose of this freely available default Healthentia-mode is two-fold. On the one hand, end-users (citizens) can use Healthentia at no cost to monitor their own health- and lifestyle parameters, and in the future benefit from the virtual coaching services offered. On the other hand, Innovation Sprint may use the data provided by end-users to do research and improve their services. For the PHYSICS use case, this default Healthentia-mode will be used, using PHYSICS components to optimize the deployment of various smart services.

A typical usage scenario for the PHYSICS use case is thus as follows:

- An individual interested in monitoring or improving their health or lifestyle will download the Healthentia mobile application to their phone.
- Before registering for a new account, the user indicates that they do not have an "invitation code" (thus entering into the *default* study).
- The user provides an email address and password and finalizes their account creation by consenting to their data being used for research purposes.
- Once in the application, the user can link their Fitbit or Garmin account to Healthentia to start providing activity and sleep data. Additionally, they can report various symptoms and events and will be able to regularly answer questionnaires related to their overall health status.

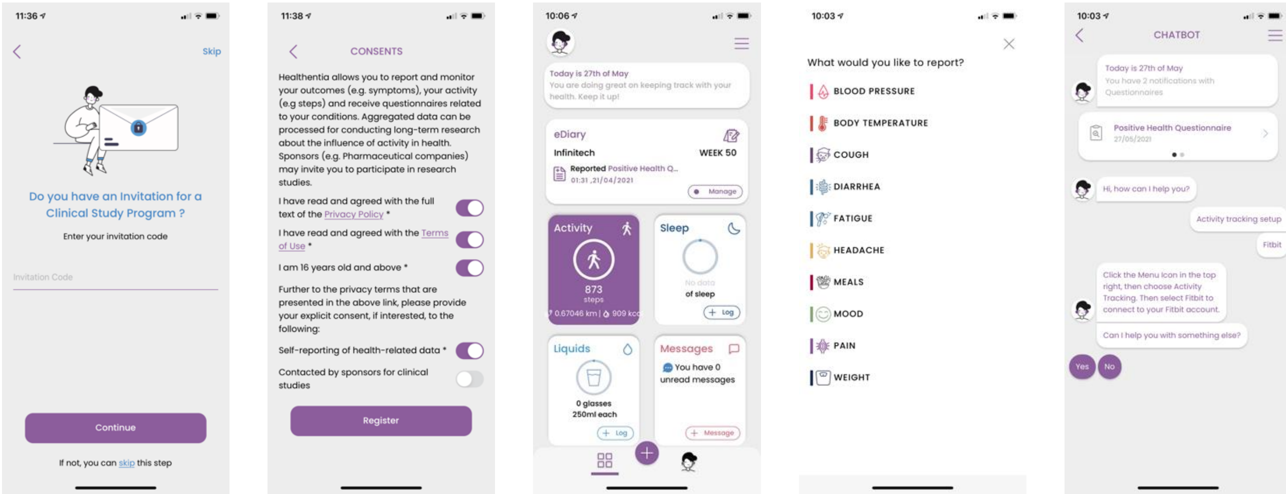

Figure 17 - Screenshots of the Healthentia mobile application.

With sufficient users having provided data for some period of time, several smart services become available. At the Healthentia Web Portal, researchers have access to three different smart services:

1. In Silico Trials.
2. Digital Composite Biomarkers.
3. Digital Phenotyping.

The data coming from users in the default Healthentia Study can be used in the following online services:

1. Create a simulated study, with generated data based on models derived from the user data.
2. Configure the usage of biomarkers to predict relevant health outcomes, such as user's daily health status.
3. Perform digital phenotyping, automatically assigning users to clusters of common observable characteristics.

There are also two offline services involved:

1. The discovery of a digital biomarker.
2. The derivation of the phenotypes via clustering and modelling the clusters.

Finally, the outcomes of the smart services may be used to provide support to the primary end-user (patient/citizen) through the virtual coach (see Figure 17– right-most image). As additional relevant data is generated through the smart services, the scripted dialogues that the virtual coach provides can be extended to include this information. A Dialogue Author can use the WOOL Editor on his personal PC to author additional dialogues for the virtual coach. These dialogues can then be uploaded to the Healthentia platform so that they become available to the end-users in the mobile application.

3.2.2 eHealth Design Scope

The main objective of the eHealth Pilot is to see how we can leverage the possibilities of the PHYSICS FaaS Platform in order to

1. Optimize system performance and scalability
2. Optimize the deployment process of new updates to the system

The expected area in which significant benefits can be obtained from using the PHYSICS platform relate to Healthentia's smart services (as described in chapter 3.2.1 above). In Table 9 below the scope of the pilot is further specified by indicating which functions could potentially be in-scope, and which functions or activities are definitely out of scope for the current pilot. Figure 18 below shows the Context Diagram of Healthentia for the PHYSICS Use Case to express the system boundaries and the actors.

Table 9 - Scope In-/Out-List for the eHealth Pilot in PHYSICS.

Item	Category	In Scope	Out of Scope
Offline training of biomarkers	R&D		X
Online prediction	Function	X	
Data Simulation	Function	X	
Deployment of AI Functions	Function	X	
Offline derivation of phenotypes	R&D		X
Online user phenotyping	Function	X	
Offline authoring of dialogues	Desktop App		X
Provisioning of Dialogues for Virtual Coach	Function	X	
Storage/Provisioning of WOOL Variables	Function	X	

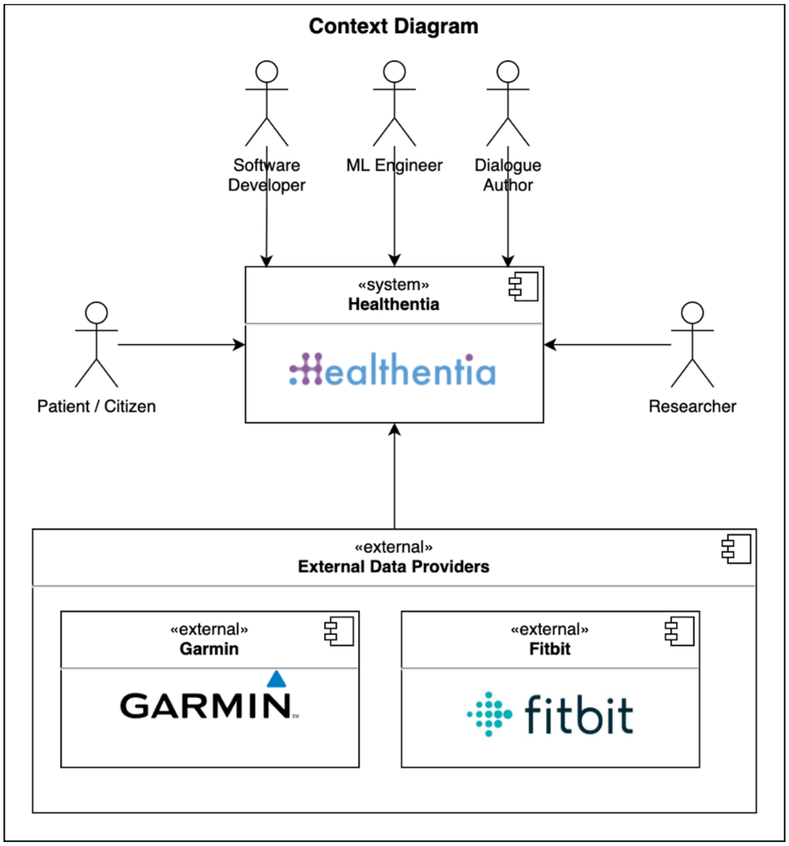

Figure 18 - User-defined context diagram for the PHYSICS eHealth Use Case.

3.2.3 eHealth Actor List

In Table 10 below, all the different actors that are relevant for the Use Case and depicted in Figure 18 above, are specified in more detail.

Table 10 - Actor List describing the 5 different Actors as depicted in the context diagram above.

Name of the role	Role Type	Characterization of the skills and the job of that actor type
Patient / Citizen	Primary actor	The Patient / Citizen uses the Healthentia mobile application to keep track of their health and to receive coaching to support achieving better health and lifestyle.
Software Developer	Secondary actor	Person who provides services related to production. No deep knowledge of FaaS.
ML Engineer	Secondary actor	Personnel of Innovation Sprint that performs the offline smart services of biomarker discovery and phenotypes' derivation.
Dialogue Author	Secondary actor	Personnel of Innovation Sprint that authors dialogues for the virtual coach, personalized using the output of the smart services.
Researcher	Primary actor	Researchers use the Healthentia Web Portal to analyze the collected data via the online services: experiment using In Silico studies, Digital Composite Biomarkers predictions, and Digital Phenotyping.
Third-Party Sensor Provider	Tertiary actor	Third-Party Sensor Providers offer data that is collected through sensors that the Patient / Citizen wear, through cloud based APIs.

3.2.4 eHealth Actor-Goal List

In Table 11 below, for each Actor defined in the Actor List of Table 10 details are provided on the specific tasks that they execute within the context of this Pilot.

Table 11 - Extended information on the use case Actors is provided in the Actor-Goal List.

Primary Actor	Task-Level Goal	Brief Description
Patient / Citizen	Monitor their physical activity patterns	The user tracks their daily physical activity through a wearable sensor. They use the Healthentia app to get insights into their physical behaviour by navigating to the Insights section of the Physical Activity page.
	Elicit advice on improving their lifestyle	After having monitored their health parameters for some time, the user engages in a conversation with the virtual coach to ask about tips and advice on how to improve his health.
Researcher	Run an in silico trial to experiment with a new trial setup	The researcher, through the web portal that has the permission to create new Studies creates a new "In Silico Study" by specifying its configuration options. An In Silico Study is a Clinical Study that uses artificial data that can be generated in various different ways, one of them being the ability to generate date from models that are created through the data collected in PHYSICS.
	Setup a Digital Composite Biomarker process to test prediction quality	The researcher enables the digital composite biomarker process that will start generating predictions on a configured outcome parameter (e.g., health status). The researcher has access to the models used and outcomes to analyze for research purposes.
	Setup a Digital Phenotyping process to perform a group analysis on collected data from Patient/Citizen users	The researcher enables the digital phenotyping process to be able to see a clustering of users (Patient/Citizen) in the default Healthentia study for research purposes.
ML engineer	Derive a new biomarker or derive the phenotypes	The ML engineer performs supervised and unsupervised ML tasks.
Software Developer	Deploy an update to an AI process	The developer wants to deploy an update to an AI process related to the digital composite biomarkers, digital phenotyping, or in silico trial data generators.
Dialogue Author	Author dialogues for the virtual coach	The dialogue author is responsible for writing content for the Virtual Coach that is available in the mobile app for the Patient/Citizen users.
Third-Party Sensor Provider	N/A	The Actor doesn't have an active role in the Use Case.

3.2.5 eHealth As-Is Architecture Views

Below, the current architecture of the Healthentia platform is described in three different views – the Domain Model View (see Figure 19), the IoT Layer View (see Figure 20), and the Deployment View (see Figure 21).

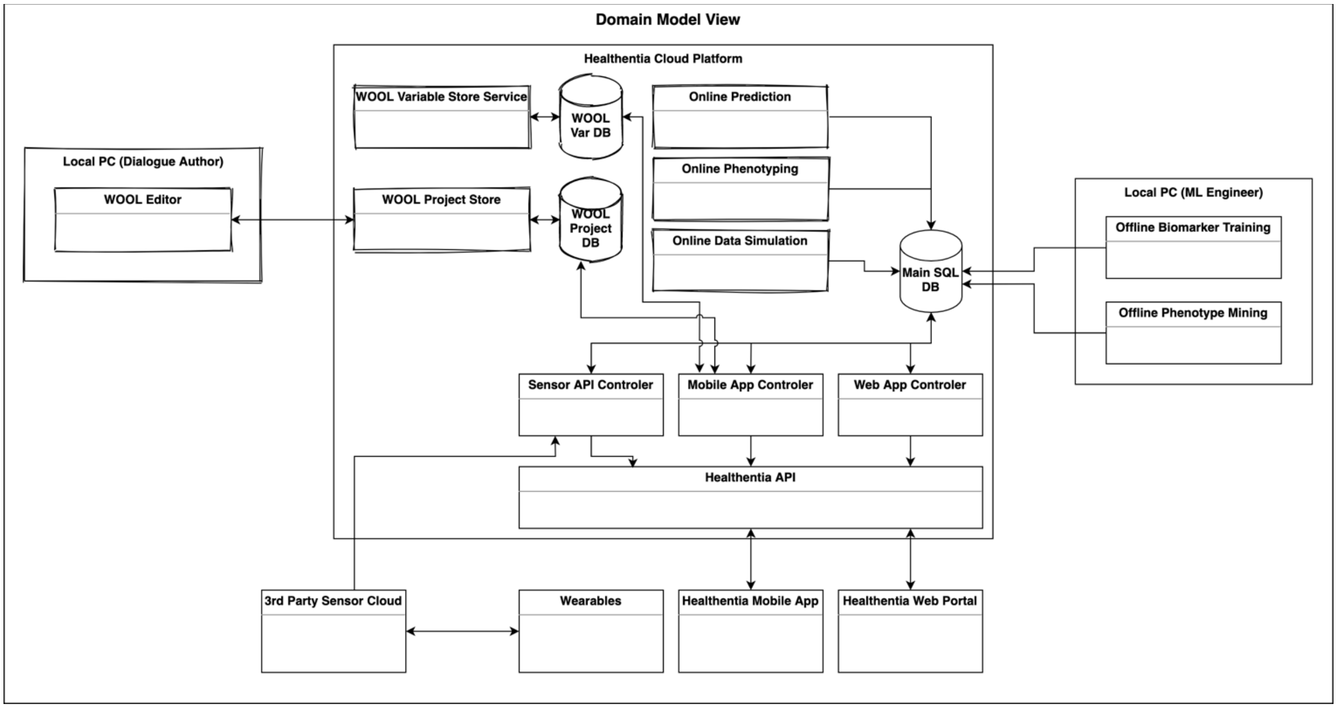

Figure 19 - Domain Model View for the Healthentia Pilot in PHYSICS.

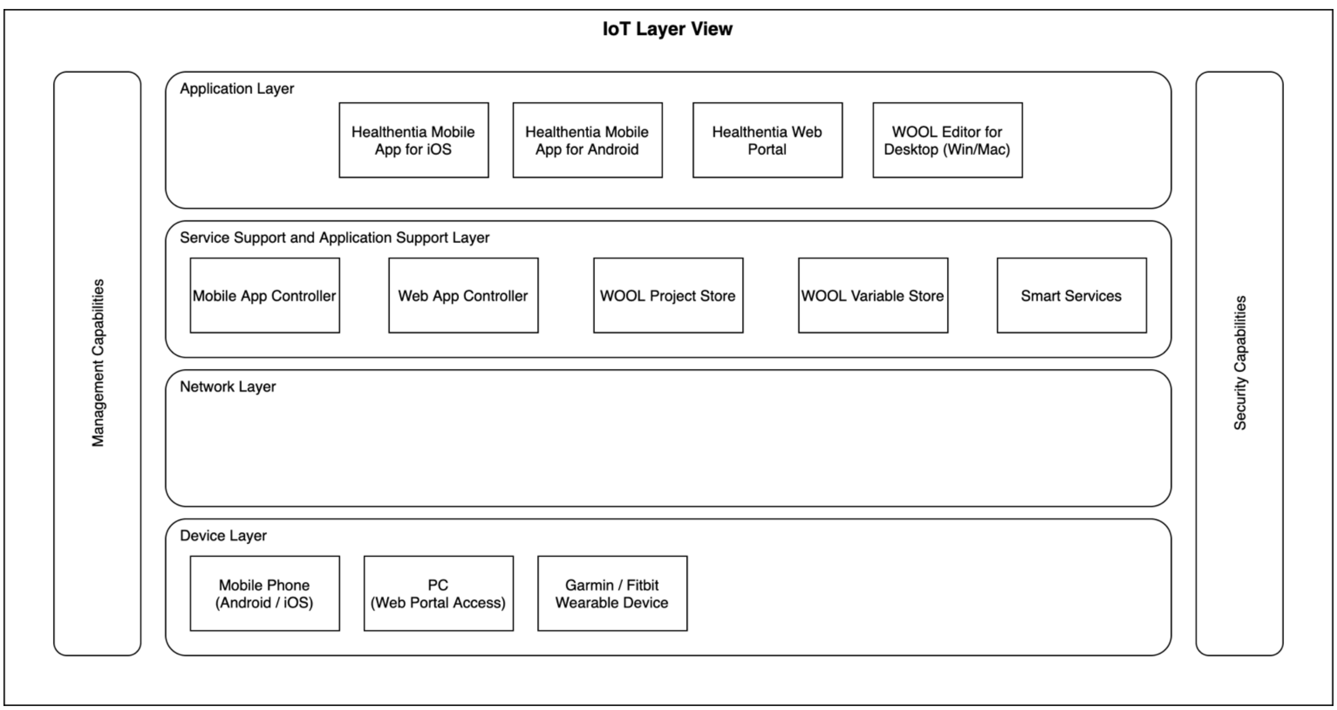

Figure 20 - IoT Layer View for the Healthentia Pilot in PHYSICS.

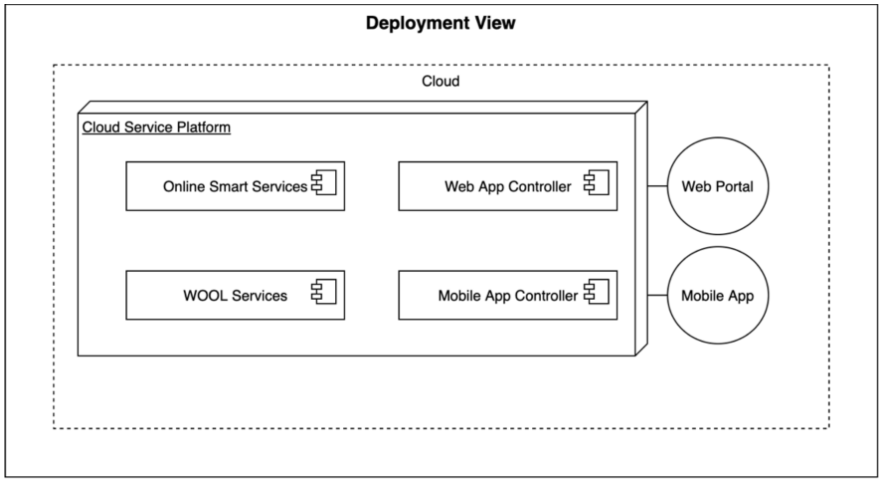

Figure 21 - Deployment View for the Healthentia Pilot in PHYSICS.

3.2.6 eHealth Actor Benefits

There are some primary actors of the eHealth Pilot that are interacting with the System that we assume will benefit fundamentally from the use of FaaS patterns and the overall PHYSICS architecture. The following Table 12 shows all identified benefits and the prioritization that was conducted during the workshop. The benefits "Automate Deployment" of the ML engineer and the benefit "Easier management of the dynamic needs of the platform" were identified as the most valuable benefits. Based on this finding these two actors were selected for the use case modeling in the following two sections.

Table 12 - Prioritization of the benefits for the primary actors with FaaS in the eHealth Pilot

Primary Actor	Benefit from FaaS	Election
ML engineer	Automate Deployment (He/she can trigger the deployment by himself/herself)	6 votes
	Rapid Pipeline Testing	3 votes
	Abstracted Development (developing the native programming language)	4 votes
Software Engineer	Easier MGMT of the dynamic needs of the platform	5 votes
	▪ Not being bothered by ML Engineers to deploy stuff ▪ Focus on main tasks	2 votes
Clinical Researcher	▪ Extension to more interactive interaction models or enables thinking about JIT-Predictions ▪ The researcher can benefit from the scalability of the services	4 votes

3.2.7 eHealth Use Case #2.1: Deployment of Service

UC2.1 is a mandatory, horizontal use case applicable for any service. It includes the experimentation with the PHYSICS Design Environment (DE) to implement services in the PHYSICS FaaS (Function-as-a-Service) way, i.e., the use inside DE of a Node-RED flow to define a function and to deploy a service.

Table 13 - eHealth Use Case #2.1: Deployment of Service

Use Case #2.1	Deployment of Service		
Brief Description	Deployment of services from Node-RED flows using the PHYSICS DE. Node-RED is used to create and locally test the flow implementing the service. The Jenkins pipeline is then invoked to build the image of the service. The image is deployed as a function using any annotation nodes in the flow, and the URL to access the service is returned.		
Context of use	An ML Engineer has finished some R&D process and is ready to deploy a new service. They use their associated Python script in the Node-RED flow, alongside more function nodes and PHYSICS patterns to handle the input and output of the script.		
Scope	PHYSICS platform		
Level	Actor level Use Case		
Primary Actor	ML Engineer		
Stakeholder & Interests	**Stakeholder**		**Interest**
	ML Engineer		In control of deployment of their own work. No dependency on Software Engineers.
	Software Engineer		Able to focus on core software engineering tasks.
	Product Owner		Fewer personnel dependencies, faster product updates.
Preconditions	Python script implementing the task at hand is created and tested		
Success End Scenario	Accessing the service from the provided URL		
Failed End Protection	New service is not deployed, ML engineer is notified by DE error messages		
Trigger	Manually triggered by primary actor		
Success Scenario	**Step**	**Action**	**System Reaction**
	1	ML Engineer designs the flow comprising nodes for functions, PHYSICS patterns and annotators	DE prompts to push to Gogs branch
	2	ML engineer tests the flow locally	DE offers debug messages in Node-RED environment
	3	ML engineer deploys service	DE provides progress indicator and upon success updates the table of services deployed from the flow (and their URLs)
	4	ML engineer tests the service	DE provides the functionality to invoke the service endpoints. Alternatively flows can be created to test the service, or some external tool like Postman can be used to post to the specific URL

Extensions	Step	Branching Action
	2a	Error: The flow does not behave as expected
	3a	Error: Jenkins notifies about a problem in creating the image and deploying it
	4a	Error: The service does not behave as expected

3.2.8 eHealth Use Case #2.2: Model inference

Model inference is the usage of an ML model with some data vectors to infer on the data, i.e., get one decision per vector. The process needs a predictive model and a set of vectors to infer upon. It is the first ML application, for which a PHYSICS service is to be built in the context of UC2.2. Like all the following UC2.3 and UC2.4, it comprises a function node at the core of its flow that is written in Python and performs the ML task at hand.

Table 14 - eHealth Use Case #2.2: Model Inference

Use Case #2.2	Model Inference		
Brief Description	A service is needed to infer on data given a model. The service has access to a pool of predictive models. The service is given some data vectors and an identifier in the pool of models. It employs the specific model to infer on the provided vectors, returning one prediction per vector.		
Context of use	An ML Engineer has trained and stored a number of predictive models. Patients generate data, from which vectors are built and are given to the model for inference. The healthcare professional receives the predictions.		
Scope	PHYSICS platform		
Level	Actor level Use Case		
Primary Actor	Healthcare professional		
Stakeholder & Interests	**Stakeholder**	**Interest**	
	ML Engineer	Provider of the models.	
	Patient	Provider of the data.	
	Healthcare professional	Consumer of the predictions.	
Preconditions	Predictive models accessible via some model identifier		
Success End Scenario	Receiving the predictions		
Failed End Protection	Predictions are not inferred. The service explains the reason via the returned error messages		
Trigger	Manually triggered by primary actor (actually a dashboard software controlled by the primary actor)		
Success Scenario	**Step**	**Action**	**System Reaction**
	1	A request is sent for inference as an action from the healthcare professional using some dashboard software	Predictions are returned
Extensions	**Step**	**Branching Action**	
	1a	Error: Unknown model identifier	
	1b	Error: Unexpected data vectors' structure	

3.2.9 eHealth Use Case #2.3: Patient phenotyping

Patient phenotyping is the process of matching vectors of data from patients to the models of different phenotypes, returning the one each vector is more likely generated from. The process needs a set of generative models and a set of vectors to match to some of them. It is the second ML application, for which a PHYSICS service is to be built in the context of UC2.3.

Table 15 - eHealth Use Case #2.3: Patient phenotyping

Use Case #2.3	Patient phenotyping		
Brief Description	A service is needed to phenotype patients. The service has access to a pool of generative models. The service is given some data vectors. It calculates the probability each vector is generated by any of the models, returning the most probable model for each vector.		
Context of use	An ML Engineer has trained and stored a number of generative models describing patient clusters (phenotypes). Patients generate data, from which vectors are built and are matched against all phenotype models to find the most likely phenotype. The healthcare professional receives the phenotypes of the patient.		
Scope	PHYSICS platform		
Level	Actor level Use Case		
Primary Actor	Healthcare professional		
Stakeholder & Interests	**Stakeholder**	**Interest**	
	ML Engineer	Provider of the models.	
	Patient	Provider of the data.	
	Healthcare professional	Consumer of the phenotypes.	
Preconditions	Accessible generative models		
Success End Scenario	Receiving the phenotypes		
Failed End Protection	Phenotypes are not estimated. The service explains the reason via the returned error messages		
Trigger	Manually triggered by primary actor (actually a dashboard software controlled by the primary actor)		
Success Scenario	**Step**	**Action**	**System Reaction**
	1	A request is sent for patient phenotyping as an action from the healthcare professional using some dashboard software	Phenotypes are returned
Extensions	**Step**	**Branching Action**	
	1a	Error: Generative models not found	
	1b	Error: Unexpected data vectors' structure	

3.2.10 eHealth Use Case #2.4: Data synthesis

Data synthesis is the process of employing a set of generative models to create synthetic data, by selecting the model to employ at every generation step based on a model transition probability matrix, returning the set of synthetic data vectors. The process needs a set of generative models, the model transition probability matrix, the number of time steps to synthesize for and the number of patients to repeat the process. It is the final ML application, for which a PHYSICS service is to be built in the context of UC2.4.

Table 16 - eHealth Use Case #2.4: Data synthesis

Use Case #2.3	Data synthesis
Brief Description	A service is needed to create synthetic data. The service has access to a pool of generative models. The service is given the number of patients to synthesize, the number of time steps to perform and the model transition probability matrix. It selects the next model to generate a vector from, generates it and accumulates all synthetic vectors in a response.

Context of use	An ML Engineer has trained and stored a number of generative models. Actual patients are used to estimate a model transition probability matrix, that can be tweaked by the study investigator to serve their goals. The study investigator receives the synthetic dataset to visualize as any actual one.		
Scope	PHYSICS platform		
Level	Actor level Use Case		
Primary Actor	Study investigator		
Stakeholder & Interests	**Stakeholder**		**Interest**
	ML Engineer		Provider of the models.
	Study investigator		Provider of the parameters of the data synthesis.
	Study investigator		Consumer of the synthetic data.
Preconditions	Accessible generative models		
Success End Scenario	Receiving the synthetic data		
Failed End Protection	Synthetic data is not produced. The service explains the reason via the returned error messages		
Trigger	Manually triggered by primary actor (a dashboard software controlled by the primary actor)		
Success Scenario	**Step**	**Action**	**System Reaction**
	1	A request is sent for data synthesis as an action from the Study investigator using some dashboard software	Synthetic data is returned
Extensions	**Step**	**Branching Action**	
	1a	Error: Generative models not found	

3.2.11 eHealth Key Performance Indicators

The following key performance indicator were defined for the eHealth pilot based on the methodology presented in chapter 2.3. The Evaluation of the KPIs in the current infrastructure has been reported in Deliverable 6.7" PHYSICS application prototypes evaluation V1" and will be updated later in the project in Deliverable 6.8 "PHYSICS application prototypes evaluation V2".

- PHYSICS DE is used to deploy a new service by an ML engineer without the need of any DevOps specialist.
- Annotation nodes in the flows can control deployment at different environments and with different settings, based on service needs.
- Response time is improved by 20% compared to the traditional deployment under bursts of small requests.
- Response time is improved by 10% when using the Request Aggregator pattern compared to the typical FaaS deployment under bursts of small requests
- Cost of maintaining the FaaS service is reduced by 30% compared to the traditional, always on deployment for the sporadic nature of the healthcare requests.

3.3 "Smart Agriculture" Pilot

3.3.1 Smart Agriculture Pilot Description

Greenhouses are nowadays the most sophisticated way to control plant environment to increase their production, reduce impact of climate uncertainty, provides physical barriers to diseases, enabling strong reduction of chemical pesticides. However, they require more and more parameters to be set by the grower (e.g., 200 in a standard soil-less glasshouse used for tomato). As a consequence, parameters are mostly set to default values, without adaptation to the location of the farm, the needs of the species and of the cultivar, their potential in yield and quality (dry matter and sugar content). In previous work on greenhouse vegetables, CybeleTech highlighted that the greenhouse modelling solutions developed could significantly improve crop management and yield estimation: Included, an economy of 50-100€/ ha/day of CO2 (92% of CO2 cost) and a reduced emission of liquid CO2 of 90% was obtained on tomato crops, yield and quality estimations on salad crops reached 95-90% of precision. This could be increased by a more connected and more reactive "digital twin" processing in quasi-RT the meteorological data of the greenhouse (hourly to daily reactions depending on the actuators). The uncertainty would be strongly reduced by automated data assimilation in the plant mechanistic model. This has already been done on field crops which does not require the management of so many management data and no need for quasi-RT answer (an hour in the greenhouse, a few days in field crops).

Data to be used by data assimilation methods consists of daily production management provided by the grower, in addition to climate data. We have to process around 30 climate variables coming every 10 minutes to 60 minutes from the greenhouse sensors monitoring temperature and humidity spatial heterogeneity. The data assimilation should need to process 500000 to 1 000 000 simulations every day on each greenhouse to manage meteorological uncertainty and correct its trajectory with existing historical data. This is an estimation as the method has never been tested on such a complex model. The models produce a lot of emergence (i.e. low input variation can have important output impact), this is very powerful to represent the biology, but more complicated to manage for data assimilation and optimal control. In addition, some modelling equations are non-continuous non derivable, so no analytical resolution can be found. Therefore, the problem is to be solved in a numerical simulation approach. A simulation runs in 1 to 5s. The storage of the output of one simulation is in the order of 10Mb. This pipeline can be parallelized through HPC (High Performance Computing) techniques.

To ensure robustness to network breakout, we need to design a system with two computation location:

- A complete version of the model resolving the problem in the Cloud on the entire greenhouse and managing climate uncertainty and spatial heterogeneity. Response should be provided into one day.
- A light version implemented in the greenhouse at an intra-hour timestamp.

3.3.2 Smart Agriculture Design Scope

The agriculture pilot aims to provide growers enhancing greenhouse management scenarios. To achieve this goal, it is necessary to have: 1) A reliable tool to gather data collected in the greenhouse; 2) Up-to-date agronomic model parametrization; 3) high performing simulation and optimization pipelines.
In Table 17 below we further specify the scope of the pilot by indicating which functions could potentially be in-scope, and which functions or activities are definitely out of scope for the current pilot. Figure 22 below shows the context diagram of the CybeleTech platform for the PHYSICS use case to express the system boundaries and the actors.

Table 17 - Scope In-/Out-List for the agriculture Use Case in PHYSICS.

Item	Category	In Scope	Out of Scope
Climate simulation	Function	X	
Plant development simulation	Function	X	
Greenhouse management optimization	Function	X	
Plant model calibration	Function	X	
Data transfer from greenhouse supervisor	Function	X	
Data visualization	Software		X
Distribution of simulations	Software	X	
Data quality control	Function	X	
Image processing	Function		X

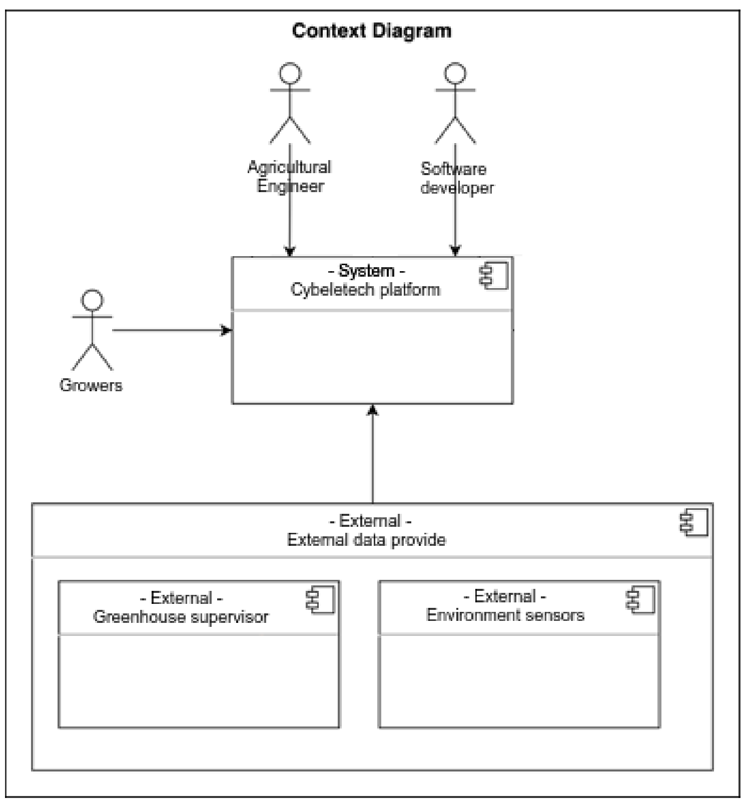

Figure 22 - Context diagram for the PHYSICS agriculture use case.

3.3.3 Smart Agriculture Actor List

In Table 18 below, all the different actors that are relevant for the use case, depicted in Figure 22 above, are specified in more detail.

Table 18 - Actor List describing the 5 different Actors as depicted in the context diagram above.

Name of the role	Role Type	Characterization of the skills and the job of that actor type
Growers	Primary actor	The growers use the CybeleTech platform to follow the environmental conditions in the greenhouse regarding plant development and generate management scenarios. They also provide information on plant development.
Software Developer	Primary actor	Person who provides services related to production. No deep knowledge of FaaS.
Agricultural Engineer	Primary actor	Personnel of CybeleTech that performs the offline climate and plant model calibration and analysis.
Greenhouse manufacturer	Secondary actor	Third-Party Providing the supervisor of the greenhouse which allow to set up environmental conditions.
Third-Party Sensor Provider	Secondary actor	Third-Party Sensor Providers offers data that is collected through sensors placed in the greenhouse, through cloud-based APIs.

3.3.4 Smart Agriculture Actor-Goal List

In Table 19 below, for each actor defined in the actor list of Table 18 details are provided on the specific tasks that they execute within the context of this pilot.

Table 19 - Extended information on the agriculture pilot actors

Primary Actor	Task-Level Goal	Brief Description
Growers	Monitoring the environmental conditions and plant development	The users use CybeleTech platform to visualize the temporal and/or spatial dynamics of environmental conditions in the greenhouse monitored by sensors. The agronomic models allow the user to follow the status of physiological variables of the plant.
	Agronomic reporting	The users upload agronomic data on the greenhouse supervisor.
	Simulation of greenhouse management scenarios	The users use CybeleTech platform to define management scenarios of the greenhouse. They can run simulation using these scenarios and the outcome in term of plant development is returned.
	Optimization of greenhouse management	The users use CybeleTech platform to explore optimal scenarios regarding one or several agronomic variables.
Agricultural Engineer	Plant model calibration	The agricultural engineer access to agronomic information made available by the growers. He run the parallelized calibration process to refine the model parametrization.
Software Developer	Deploy an update to simulation process	The developer wants to deploy an update to the simulation process related to the climate or agronomic models.
	New greenhouse context adaption	The developer wants to deploy the pipeline for the new greenhouse and potentially update the routines.

3.3.5 Smart Agriculture Architecture View

Below, the current architecture of the CybeleTech Smart Agriculture platform is depicted in two different views – the Domain Model View in Figure 23 and the Deployment View in Figure 24.

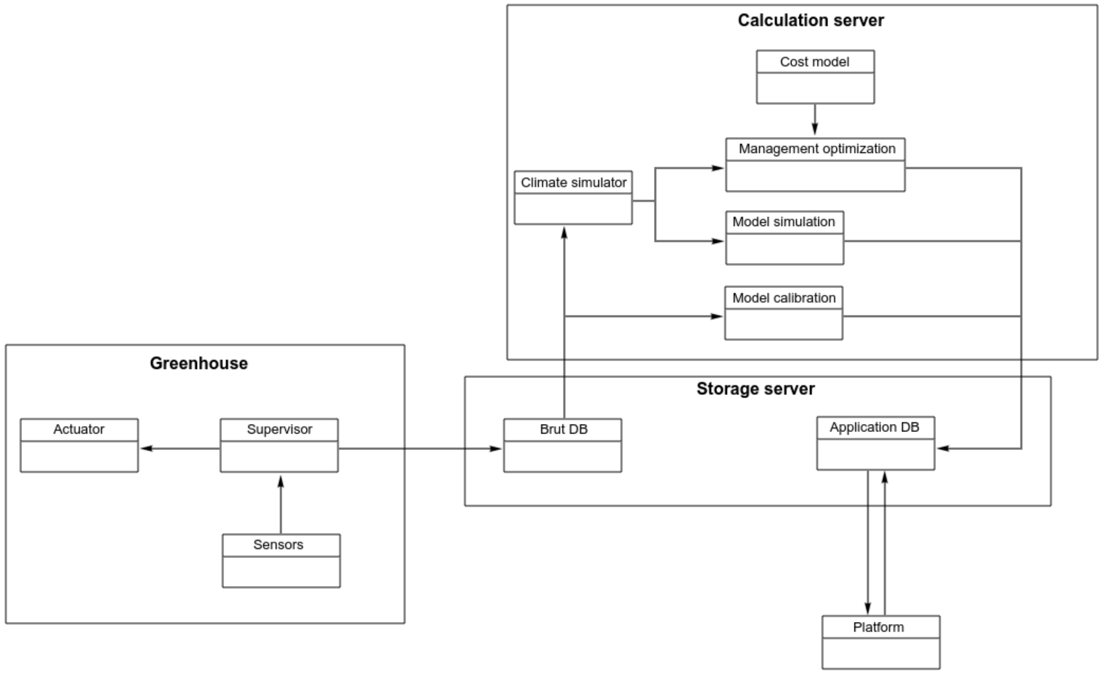

Figure 23 - Domain Model View for the agriculture Pilot in PHYSICS.

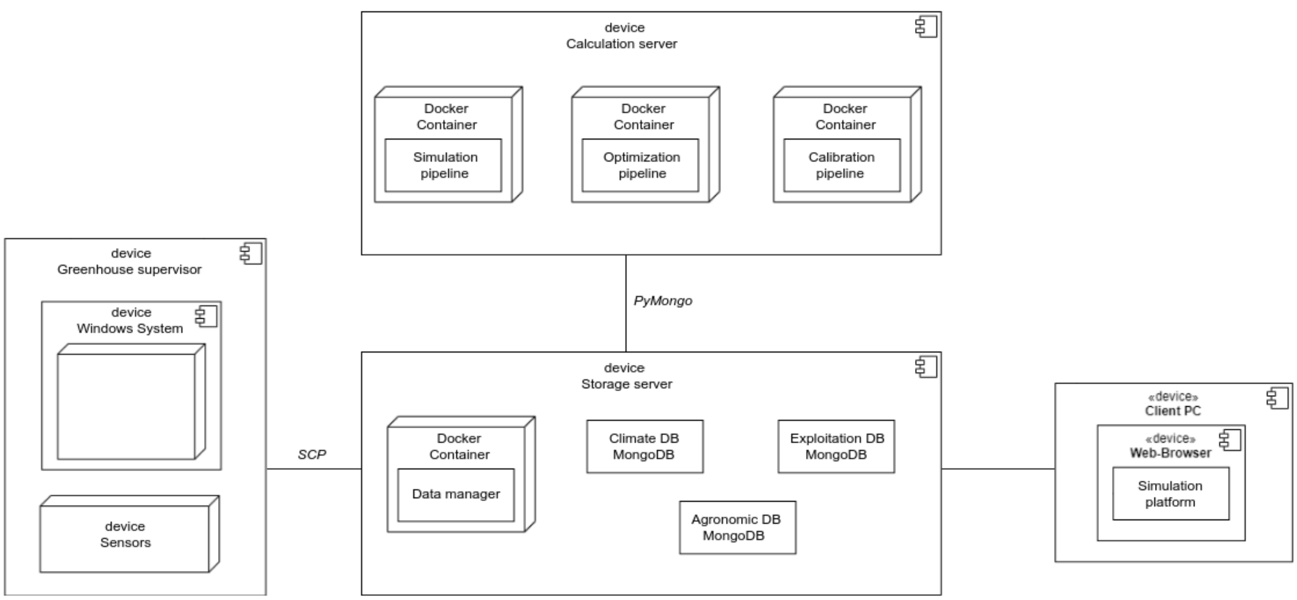

Figure 24 - Deployment View for the agriculture Pilot in PHYSICS.

3.3.6 Smart Agriculture Actor Benefits

There are some primary actors of the Smart Agriculture Pilot that are interacting with the system that is assumed to benefit fundamentally from the use of FaaS patterns and the overall PHYSICS architecture. The following Table 20 shows all identified benefits and the prioritization that was conducted during the workshop. The benefit "flexibility in data retrieval and ingestion or adaption" was identified as the most valuable benefit. The Software Developer has two identified use cases according to the Actor-Goal List that are described in the following sections. The other primary actor that was selected in the prioritization was the agriculture engineer which has one use case that is described in section 3.3.9.

Table 20 - Prioritization of the benefits for the primary actors with FaaS in the Smart Agriculture Pilot

Primary Actor	Benefit from FaaS	Election
Growers	Gain in cost	2 votes
	Continuity of services	4 votes
Software Engineer	Good practices of development	2 votes
	Reusability of the functions defined in FaaS	2 votes
	▪ Flexibility in data retrieval and ingestion or adaptation due to Node-RED / ▪ Graphical environment / ▪ Ability to include arbitrary flows that may extend logic (e.g., for increase of reliability and data retrieval in intermittent failures)	5 votes
Agriculture Engineer	Event based and scalable triggering of calibration	3 votes + Joker
Business Manager	Improved cost control	Added after voting

3.3.7 Smart Agriculture Use Case #3.1: Deploy an update to simulation process

The first use case that was identified as most relevant to implement within PHYSICS was the use case "Deploy an update to simulation process". The use case was chosen from the previously benefit prioritization of the primary actors. Figure 25 shows the activity diagram of the simulation use case with the currently used architecture and the Table 21 below shows the scenario of the first Smart Agriculture use case.

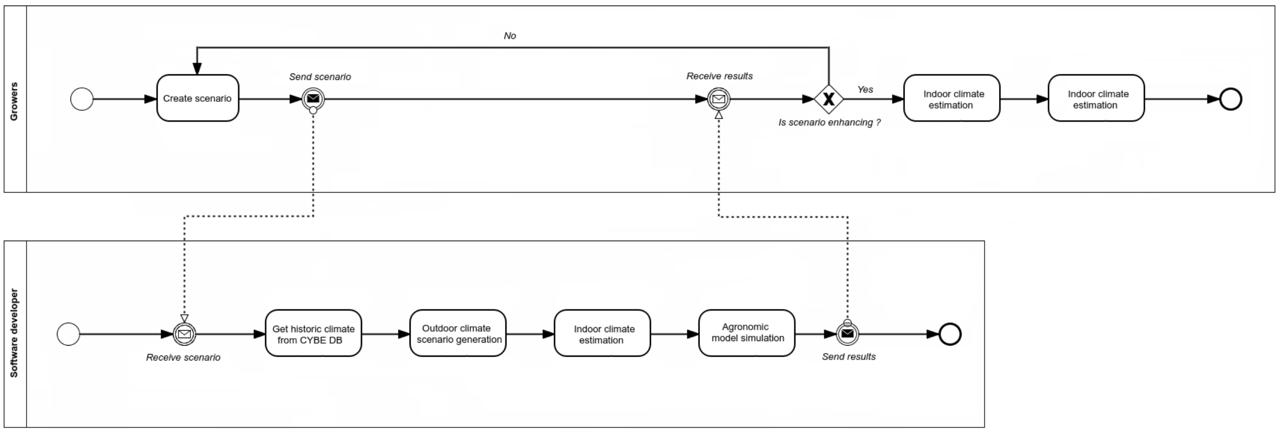

Figure 25 - Smart Agriculture Use Case #3.1 AS-IS BPMN Diagram

Table 21 - Smart Agriculture Use Case #3.1: Deploy an update to simulation process

Use Case #3.1	**Deploy an update to simulation process**	
Brief Description	The growers wants to evaluate the performance of a management scenario according to plant production. They send a request to the system with the management scenario as input. The system simulates indoor climate and its impact on the plant and return the results.	
Context of use	Occurs upon the growers' demand to have an update deployed.	
Scope	CybeleTech smart agriculture platform / PHYSICS platform.	
Level	System Use Case	
Primary Actor	Software Developer	
Stakeholder & Interests	**Stakeholder**	**Interest**
	Software engineer	Ease the deployment of updates
	Software engineer	Ease the deployment in new greenhouse.
	Agronomic engineer	Ease the decision-making regarding model calibration.
	Growers	Reduce the costs
	Growers	Enhance the robustness
Preconditions	Sensors have been installed to retrieve their status and information, Data flow has been designed and Function or Service has been designed on paper	
Success End Scenario	Deployment updates a data retriever at the edge	
Failed End Protection	-	
Trigger	▪ Software Developer suggest update ▪ Growers ask for update	

Success Scenario	Step	Action	System Reaction
	1	Software Developer designs flow to access and aggregate data	Display patterns, suggestions and available nodes in the design environment
	2	The actor includes nodes and patterns in flow	Local Testing Environment
	3	Developer parameterizes patterns	System may undertake retrieval of datasets necessary for parameterization
	4	Include existing components as functions	Package Docker Image as function
	5	Create application workflow	Extract workflow structure and register in platform
	6	Use annotations to dictate requirements of edge placement	Pass annotations to platform layer
	7	Validate the service on experimental data	Create deployable artefact of the flow (System deploys functionality in dev environment)
	8	Developer releases the flow	System submits the flow

Figure 26 shows the activity diagram of the use case that the future architecture should support. The diagram is supporting the Use Case text of Table 21.

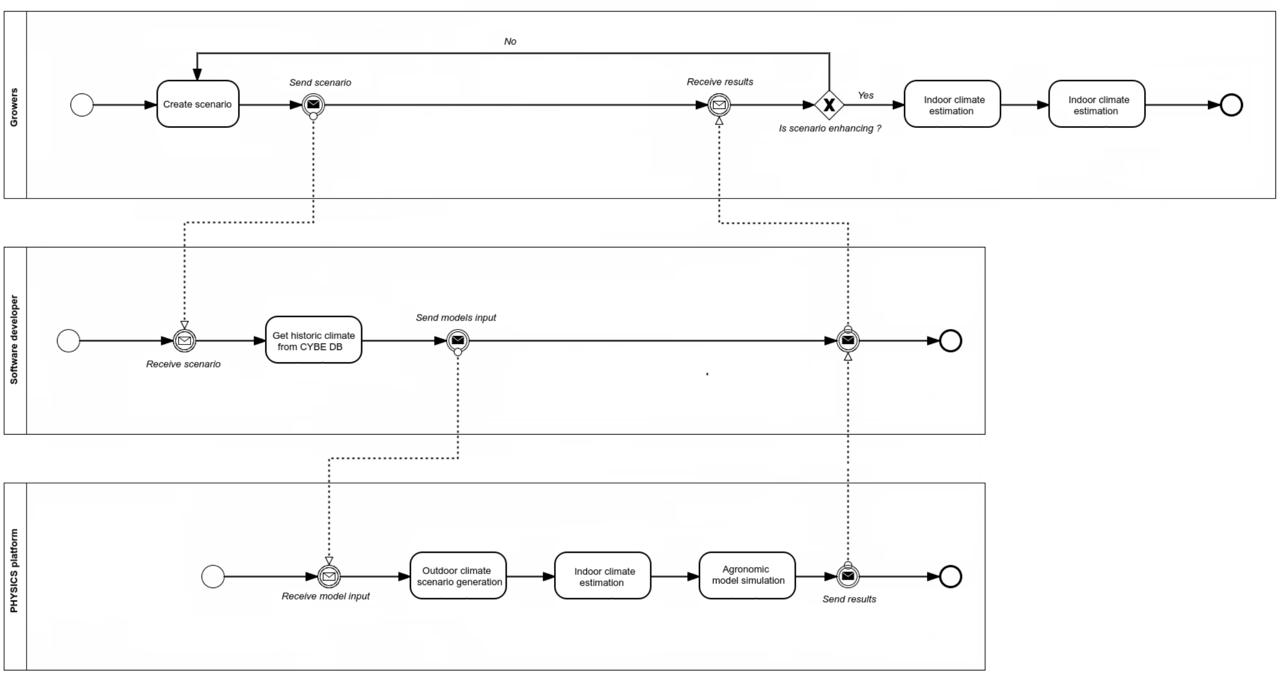

Figure 26 - Smart Agriculture Use Case #3.1 TO-BE BPMN Diagram

3.3.8 Smart Agriculture Use Case #3.2: New greenhouse context adaption

The second use case that was identified as relevant to implement within PHYSICS was the use case "New greenhouse context adaption". The use case was chosen from the previously benefit prioritization of the primary actors. Figure 27 shows the activity diagram of the use case with the currently used architecture and the Table 22 below shows the scenario of the third Smart Agriculture use case.

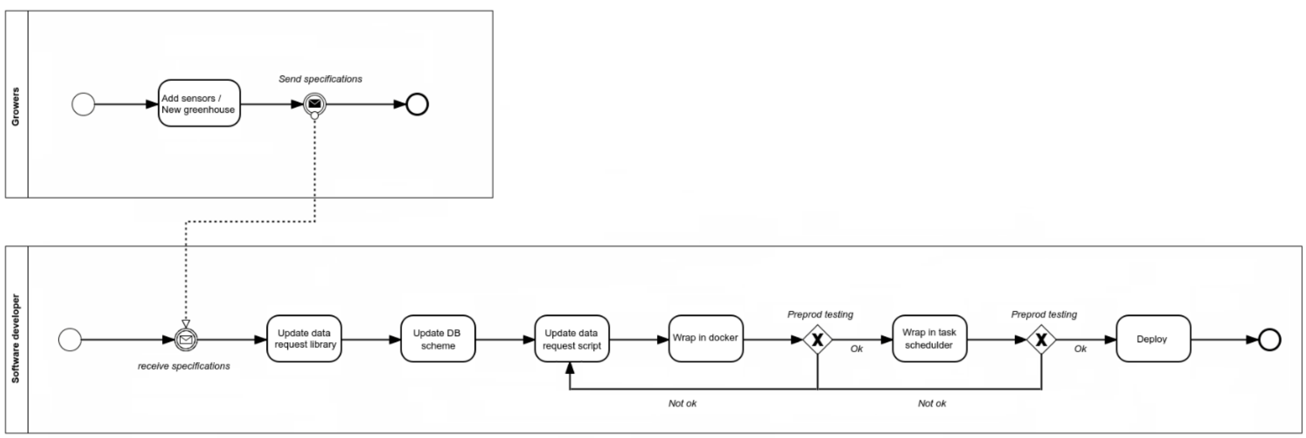

Figure 27 - Smart Agriculture Use Case #3.2 AS-IS BPMN Diagram

Table 22 - Smart Agriculture Use Case #3.2: New greenhouse context adaption

Use Case #3.2	**New greenhouse context adaption**	
Brief Description	The grower wants to deploy the system in a greenhouse with different sensors and/ or supervisor system. He notifies CybeleTech. The software engineer deploys the pipeline for the new greenhouse and potentially updates the routines.	
Context of use	The UC always occurs when new sensors are needed or when the system is deployed in a new greenhouse.	
Scope	CybeleTech smart agriculture platform / PHYSICS platform	
Level	System Use Case	
Primary Actor	Software Developer	
Stakeholder & Interests	**Stakeholder**	**Interest**
	Software engineer	Ease the deployment of updates
	Software engineer	Ease the deployment in new greenhouse
	Agronomic engineer	Ease the decision-making regarding model calibration
	Growers	Reduce the costs
	Growers	Enhance the robustness
Preconditions	Sensors have been installed to retrieve their status and information, Data flow has been designed and Function or Service has been designed on paper	
Success End Scenario	Deployment of a Data Ingestion Process	
Failed End Protection	-	
Trigger	Manually triggered notification from the grower about new specifications of newly added sensors or of a newly added greenhouse	

Success Scenario	Step	Action	System Reaction
	1	Software Developer designs flow to access and aggregate data	Display patterns, suggestions and available nodes in the design environment
	2	The actor includes reliability patterns for data ingestion and data adaption patterns for data models.	Local Testing Environment
	3	Developer parameterizes patterns	System may undertake retrieval of datasets necessary for parameterization
	4	Include existing components as functions or service	Package Docker Image as function
	5	Create application workflow	Extract workflow structure and register in platform
	6	Use annotations to dictate requirements of edge placement	Pass annotations to platform layer
	7	Validate the service on experimental data	Create deployable artefact of the flow (System deploys functionality in dev environment)
	8	Developer releases the flow	System submits the flow

Figure 28 shows the activity diagram of the use case that the future architecture should support. The diagram is supporting the use case text of Table 22.

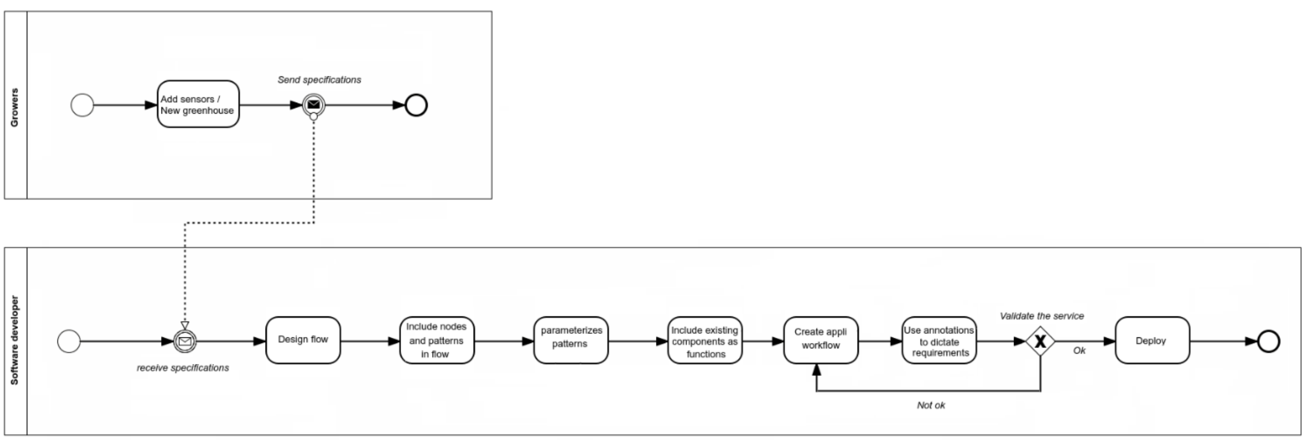

Figure 28 - Smart Agriculture Use Case #3.2 TO-BE BPMN Diagram

3.3.9 Smart Agriculture Use Case #3.3: Deploy a calibration

The third use case that was identified to implement within PHYSICS was the use case “Deploy a calibration”. The use case was also chosen from the previously benefit prioritization of the primary actors. Figure 29 shows the activity diagram of the “Calibration” use case with the currently used architecture and the Table 23 below shows the scenario of the third Smart Agriculture use case.

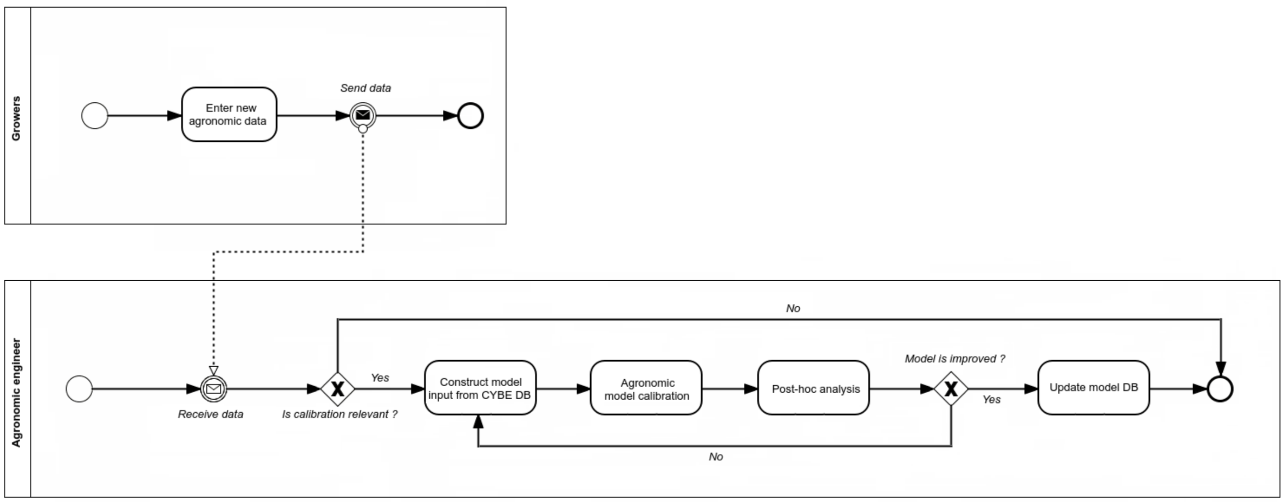

Figure 29 - Smart Agriculture Use Case #3.3 AS-IS BPMN Diagram

Table 23 - Smart Agriculture Use Case #3.3: Deploy a calibration

Use Case #3.3	**Deploy a calibration**	
Brief Description	The grower has performed new measurements on plants. He saves the information using the supervisor. The agriculture engineer evaluates the relevance of performing a new calibration according to the data. If the relevant calibration is run, evaluated and if it improves the simulation results, the new parameter set is saved.	
Context of use	The calibration UC occurs when new agronomic data becomes available.	
Scope	CybeleTech smart agriculture platform / PHYSICS platform	
Level	System Use Case	
Primary Actor	Agronomic Engineer	
Stakeholder & Interests	**Stakeholder**	**Interest**
	Software engineer	Ease the deployment of updates
	Software engineer	Ease the deployment in new greenhouse
	Agronomic engineer	Ease the decision-making regarding model calibration
	Growers	Reduce the costs
	Growers	Enhance the robustness
Preconditions	Sensors have been installed to retrieve their status and information, Data flow has been designed and Function or Service has been designed on paper	
Success End Scenario	Deploy an optimization process	
Failed End Protection	-	
Trigger	Manually triggered notification from the grower about new agronomic data	

Success Scenario	Step	Action	System Reaction
	1	Software Developer designs flow to access and aggregate data	Display patterns, suggestions and available nodes in the design environment
	2	Parallel Synchronization Patterns	Local Testing Environment
	3	Developer parameterizes patterns	System may undertake retrieval of datasets necessary for parameterization
	4	Execute optimization function component	Package Docker Image as function
	5	Create application workflow	Extract workflow structure and register in platform
	6	Placement on the Cloud, QoS	Pass annotations to platform layer
	7	Validate the service on experimental data	Create deployable artefact of the flow (System deploys functionality in dev environment)
	8	Developer releases the flow	System submits the flow

Figure 30 shows the activity diagram of the use case that the future architecture should support. The diagram is supporting the use case text of Table 23.

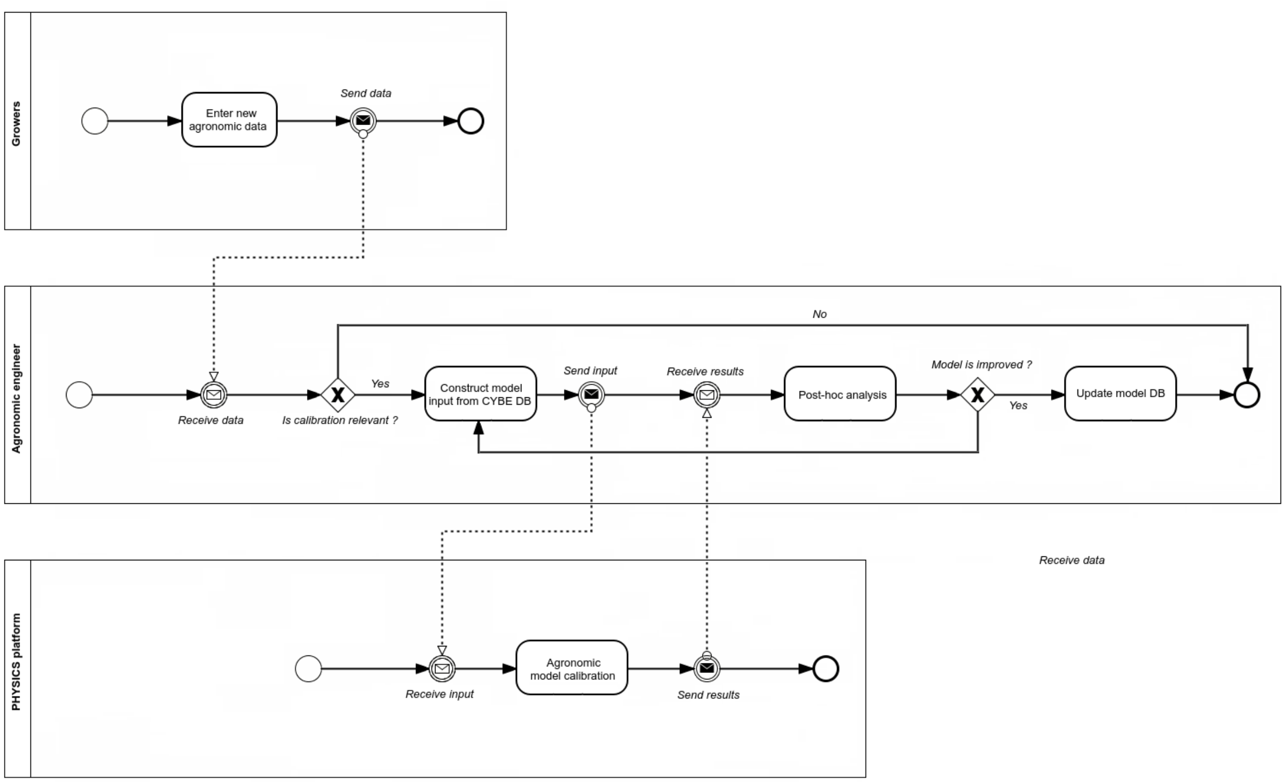

Figure 30 - Smart Agriculture Use Case #3.3 TO-BE BPMN Diagram

3.3.10 Smart Agriculture Use Case #3.4: Scaling up

A fourth use case has been identified during the first period: “Scaling up”. This use case aims at demonstrating how PHYSICS platform might contribute to the industrialization of Cybeletech greenhouse solution with the objective of easing the business process while ensuring a consistent service quality. The Table 24 below shows the scenario of the third Smart Agriculture use case.

Table 24 - Smart Agriculture Use Case #3.4: Scaling up

Use Case #3.4	Scaling up		
Brief Description	The number of growers using the solution, and then the number of requests for simulation and optimization, is increasing. The response time of the system must remain the same with no increase of cost for the growers or for the company.		
Context of use	The scaling up UC occurs when several new growers want to benefit from the solution.		
Scope	CybeleTech smart agriculture platform / PHYSICS platform		
Level	System Use Case		
Primary Actor	Business manager		
Stakeholder & Interests	**Stakeholder**		**Interest**
	Business manager		Ease the business plan structuration
	Business manager		Ease the billing process
	Software Engineer		Reduce infrastructure management effort
Preconditions	Simulation and optimization pipelines as FaaS have been implemented and deployed.		
Success End Scenario	The time needed to process growers’ requests remains the same as the number of growers increases. Both the cost for growers and the company income by grower remain stables.		
Failed End Protection	-		
Trigger	New growers want to access the greenhouse solution.		
Success Scenario	**Step**	**Action**	**System Reaction**
	1	Business manager defines scaling up scenarios.	-
	2	Software Engineer deploys the solution as FaaS	System deploys functionality in production environment
	3	Software Engineer simulates load increase by sending multiple requests	Simulation / optimization pipelines are run as FaaS
	4	Software Engineer monitors the response time with PHYSICS tools	Response time remains stable while load increases
	5	Business manager makes cost projection according to cloud fees.	Billing by growers is easy to produce and the unitary cost remain stable

3.3.11 Smart Agriculture Key Performance Indicators

For each use case several objectives have been defined to evaluate the performances improvements achieved through PHYSICS platform and projects products.

With UC #3.1 we seek to evaluate the gain in robustness and the reduction in the effort to deploy the solution induced by the integration of PHYSICS components. The associated KPIs are the following:

- **KPI01 – Amount of data lost** in relation to the number of connection failure measures the reliability of the data collection pipeline
- **KPI02 – Deployment time** is the sum of time needed to adapt the production environment to the greenhouse infrastructure specificities and to adapt the data collection procedure to greenhouse sensors and agronomic properties.

In UC #3.2 the objective is to implement and run the simulation pipeline as FaaS. This step is a prerequisite for UC #3.3 and #3.4. The main benefits expected are the reduction of maintenance costs for Cybeletech through the reduction of on-premises servers and a better mapping between the need of the growers and the fees. Those are direct benefit of FaaSification and the main concern is then:

- **KPI03- Effort to adapt the simulation pipeline** to FaaS context.

With UC #3.3 we seek to evaluate the gain in performance induced using PHYSICS platforms with:

- **KPI04 – Response time** of the optimization pipeline with increasing number of scenarios explored.

With UC #3.4 we will explore the scaling-up facilities offered by PHYSICS platform in relation to the unitary cost of functions. The associated

- **KPI 05 – Response time** of the simulation / optimization pipelines with increasing number of requests.

4 CONCLUSION AND NEXT STEPS

This documentation showed the analysis of the three pilots and how the systems of the different pilots are defined, and delimited, which actors use the pilots' system and which actors directly benefit from the PHYSICS platform. This second version of this document provides the method for designing and selecting appropriate KPIs that allow measuring the success of architectural progress in the rest of the project during experimentation and evaluation. Likewise, use cases were adapted, replaced, or supplemented by previous findings and experiments. In the course of the further project, the use cases will be continuously adapted to new requirements in Task 6.3 "Use Cases Adaptation & Experimentation". Orientation will be ensured by measuring the pilot related KPIs to be measured in the evaluation phase in Task 6.4 "Use Cases Evaluation".

5 APPENDIX

5.1 Use Case Modelling Workshops Appendix

5.1.1 Workshop Team Pictures

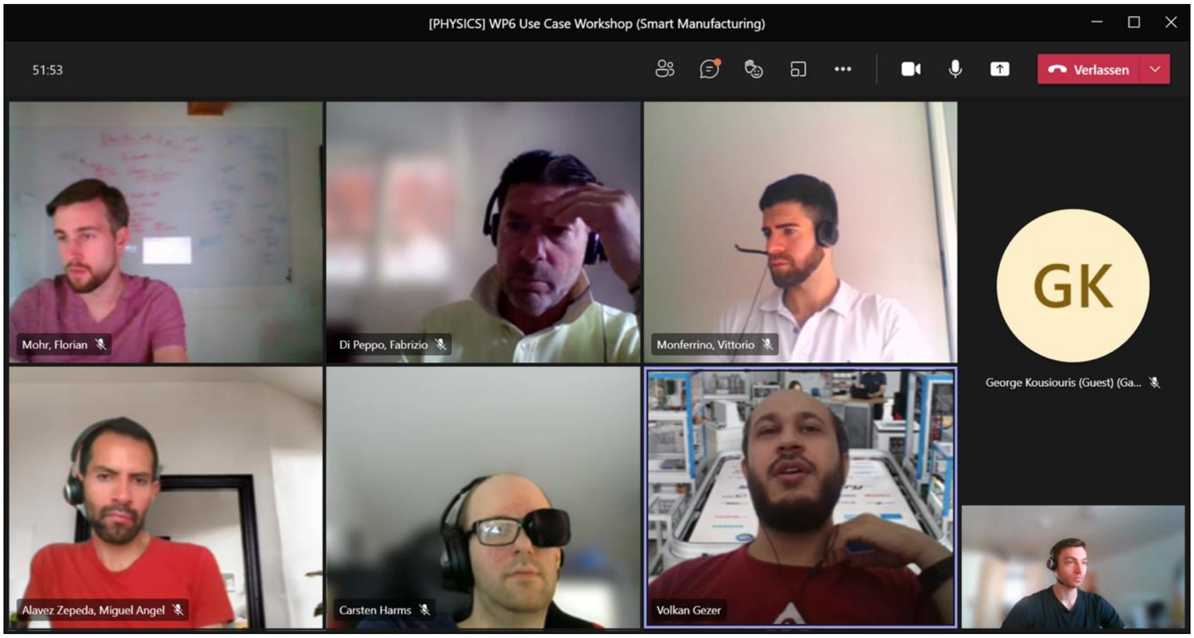

Figure 31 - Team picture of the Smart Manufacturing Use Case Workshop

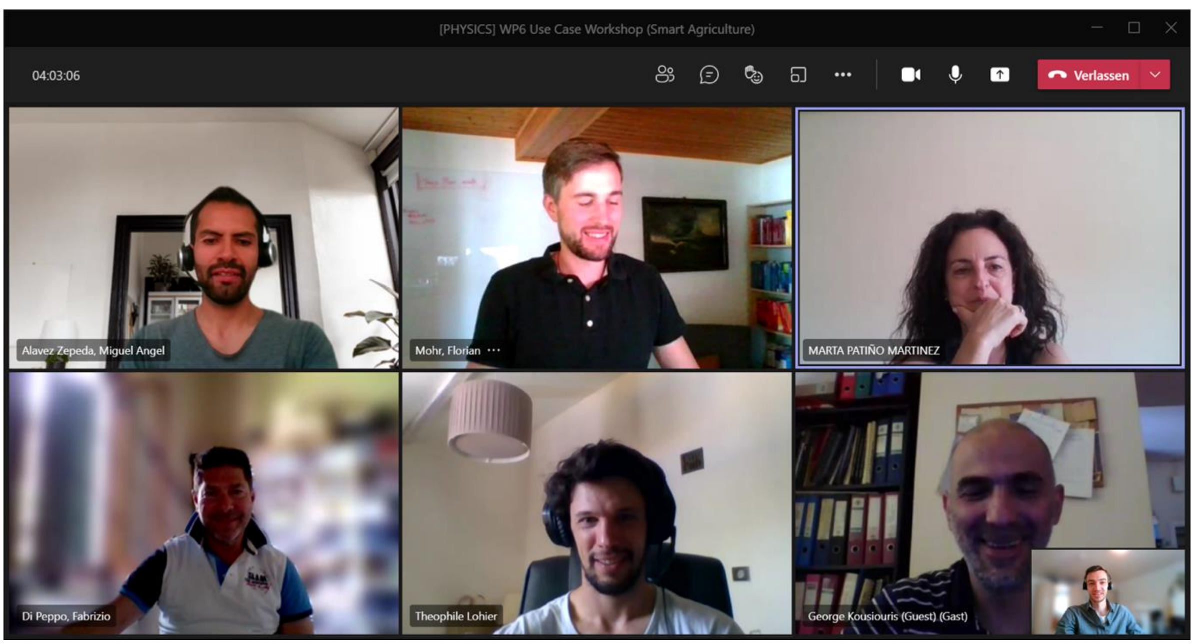

Figure 32 - Team picture of the Smart Agriculture Use Case Workshop

5.1.2 Smart Manufacturing Workshop Screenshots

The following screenshot in Figure 33 shows the quotes from all workshop participants as they would measure the personal success of the PHYSIK project.

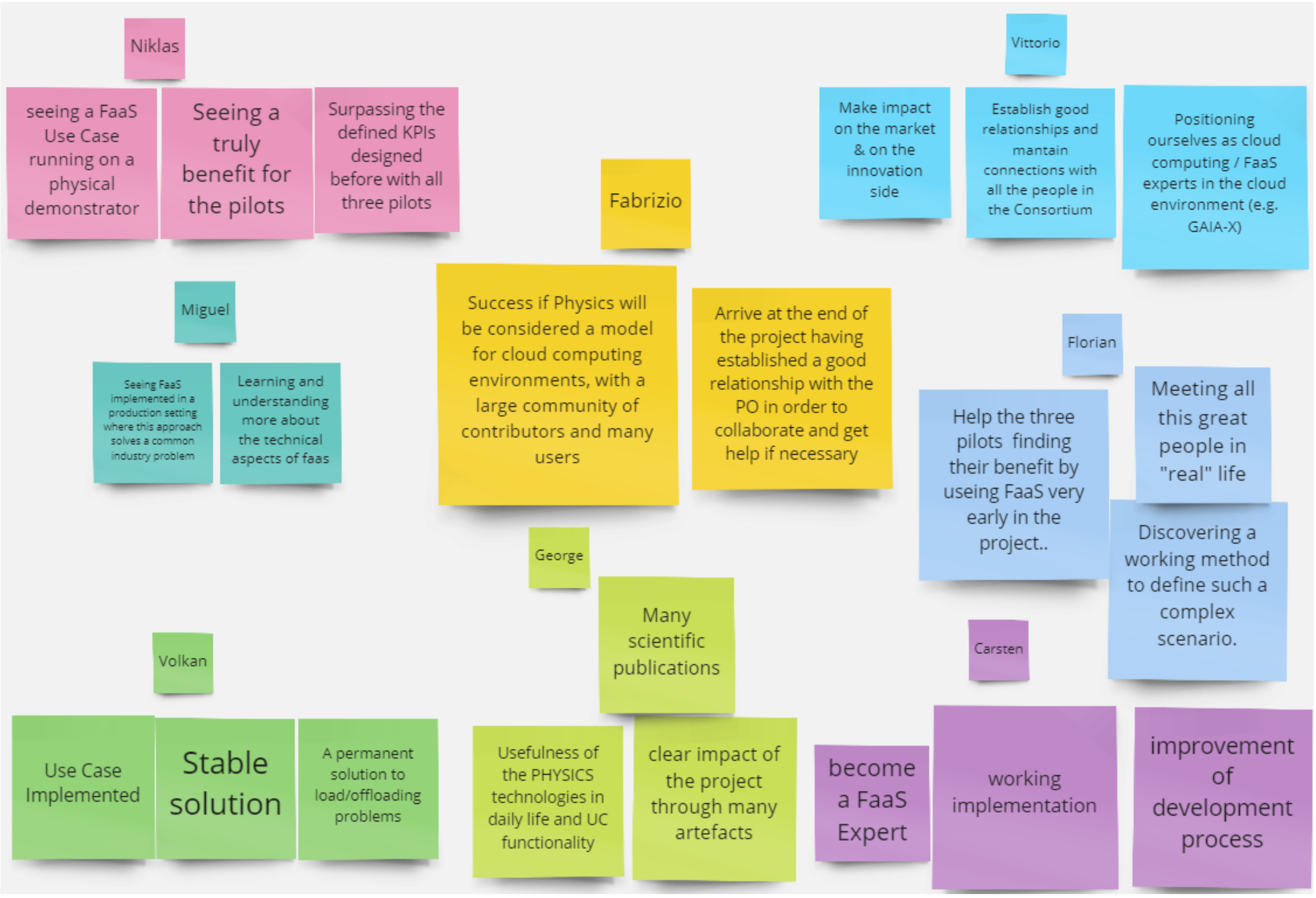

Figure 33 - Personal success of Smart Manufacturing Workshop participants

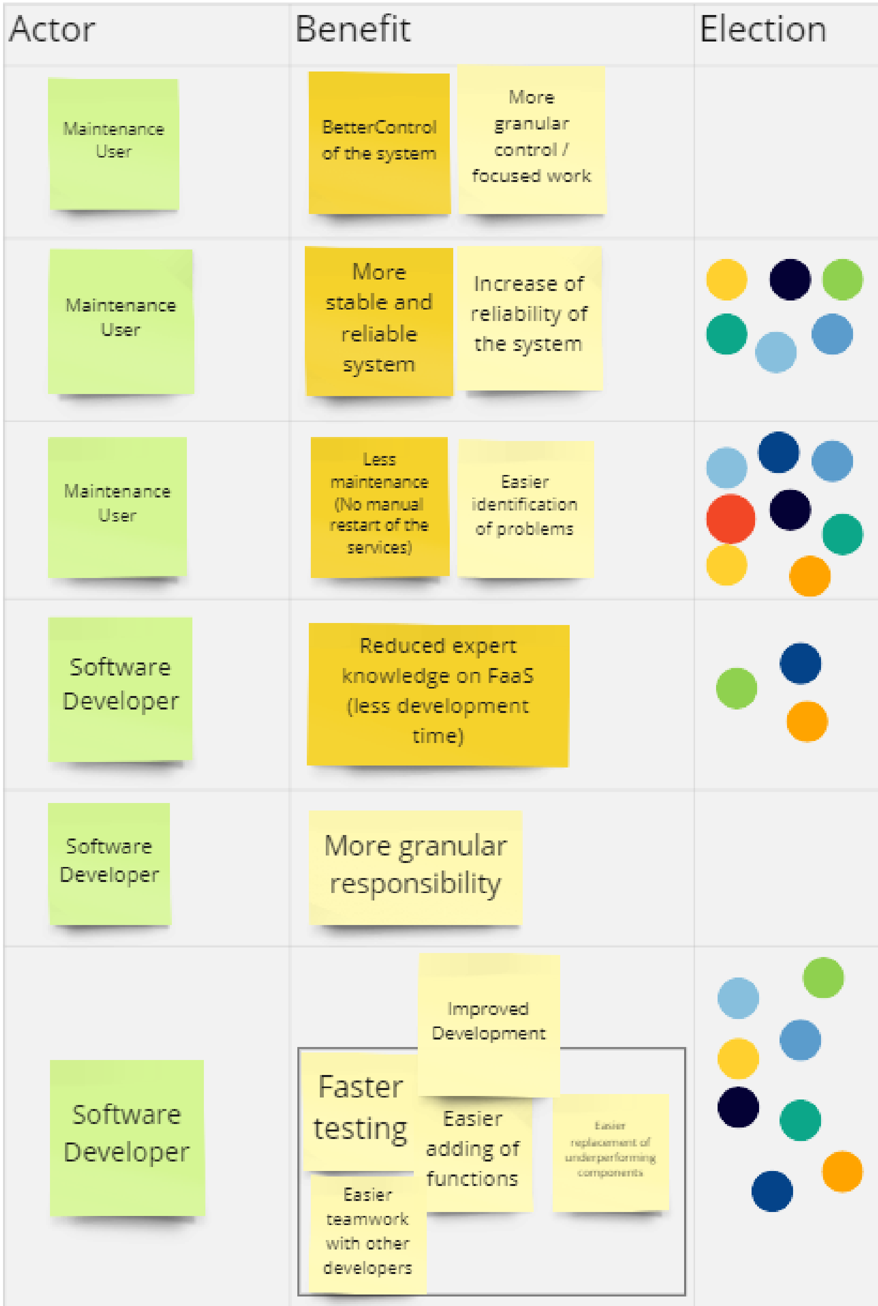

Figure 34 - Actor benefit prioritization of the Smart Manufacturing Pilot

The following Figure 35 shows how the use case scenario was developed around a primary actor and its goal to achieve in the interaction with the system.

Figure 35 - Development of the Smart Manufacturing Use Case #1.1

5.1.3 eHealth Workshop Screenshots

Figure 36 - Personal Success of eHealth Workshop participants

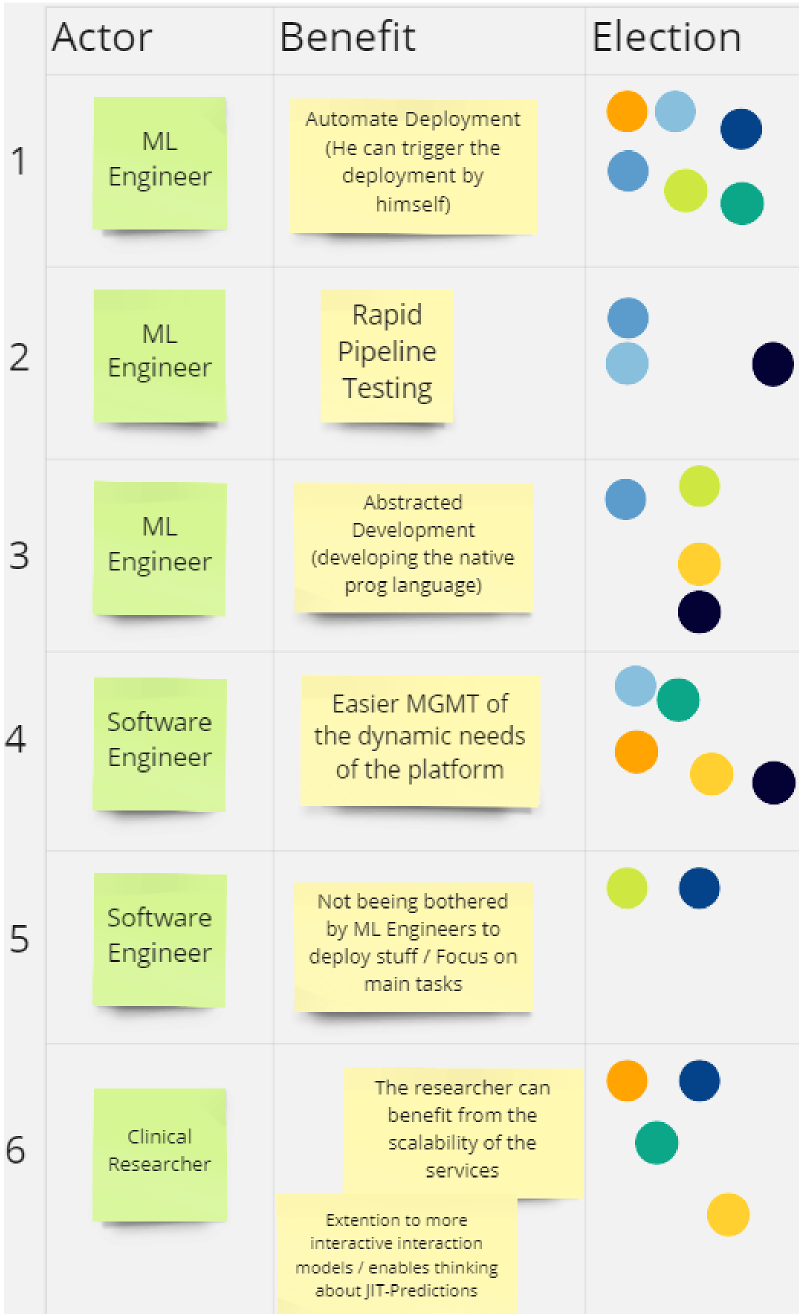

Figure 37 - Actor benefit prioritization of the eHealth Pilot

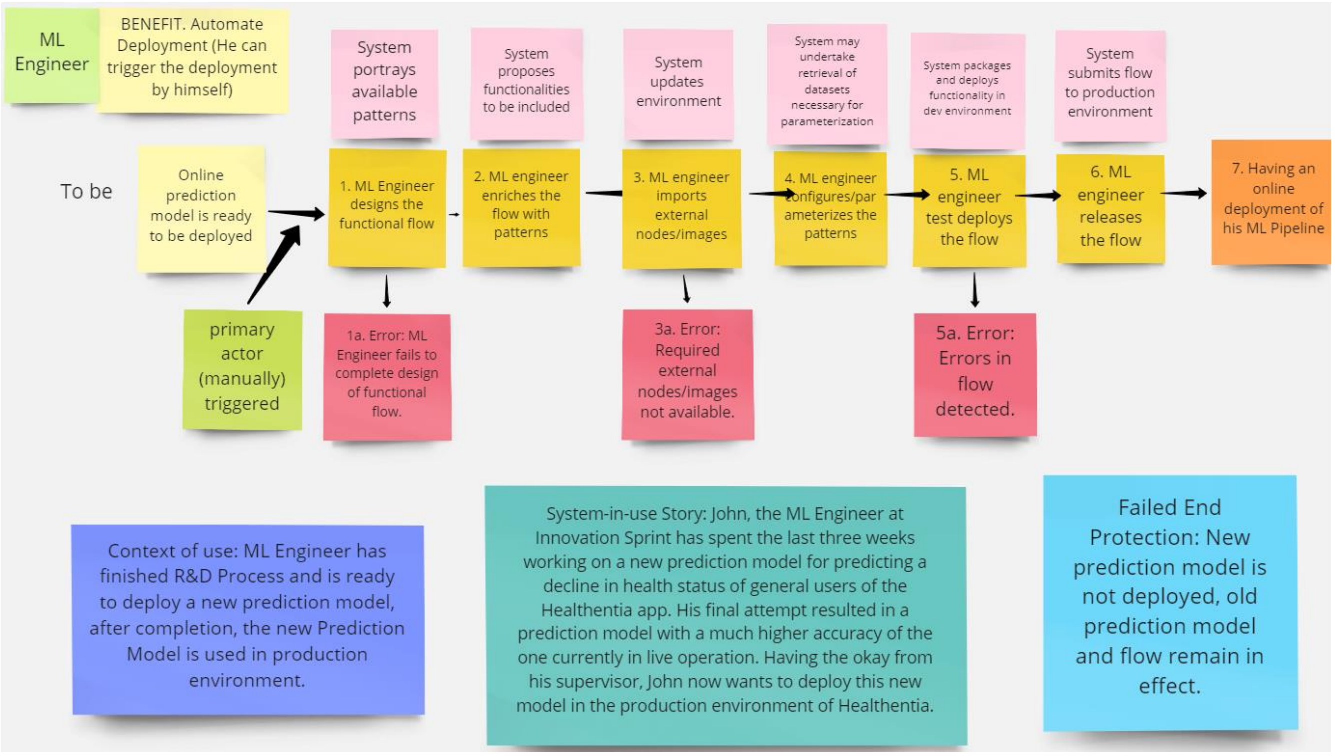

Figure 38 - Development of the eHealth Use Case #2.1

5.1.4 Smart Agriculture Workshop Screenshots

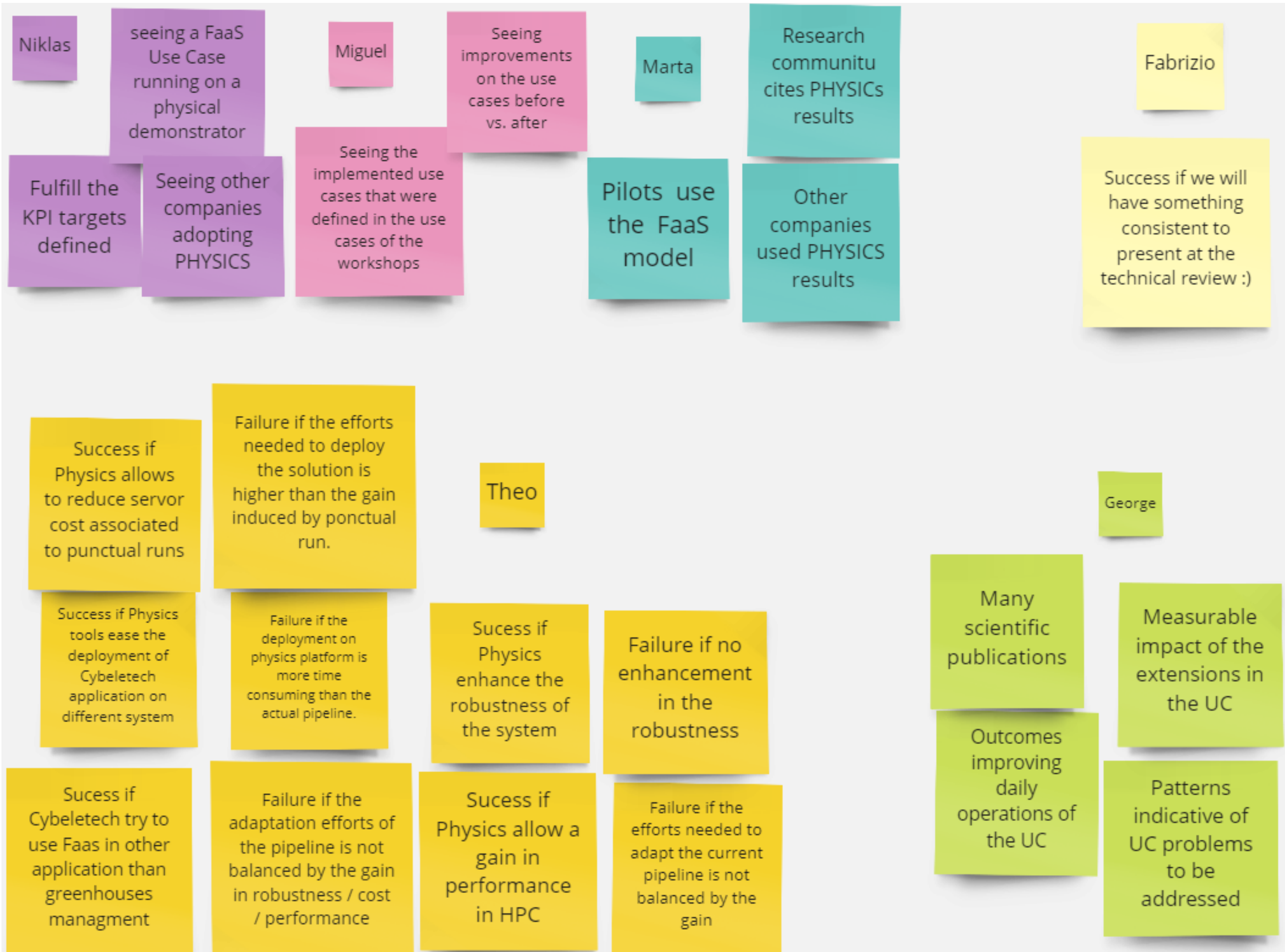

Figure 39 - Personal Success of Smart Agriculture Workshop participants

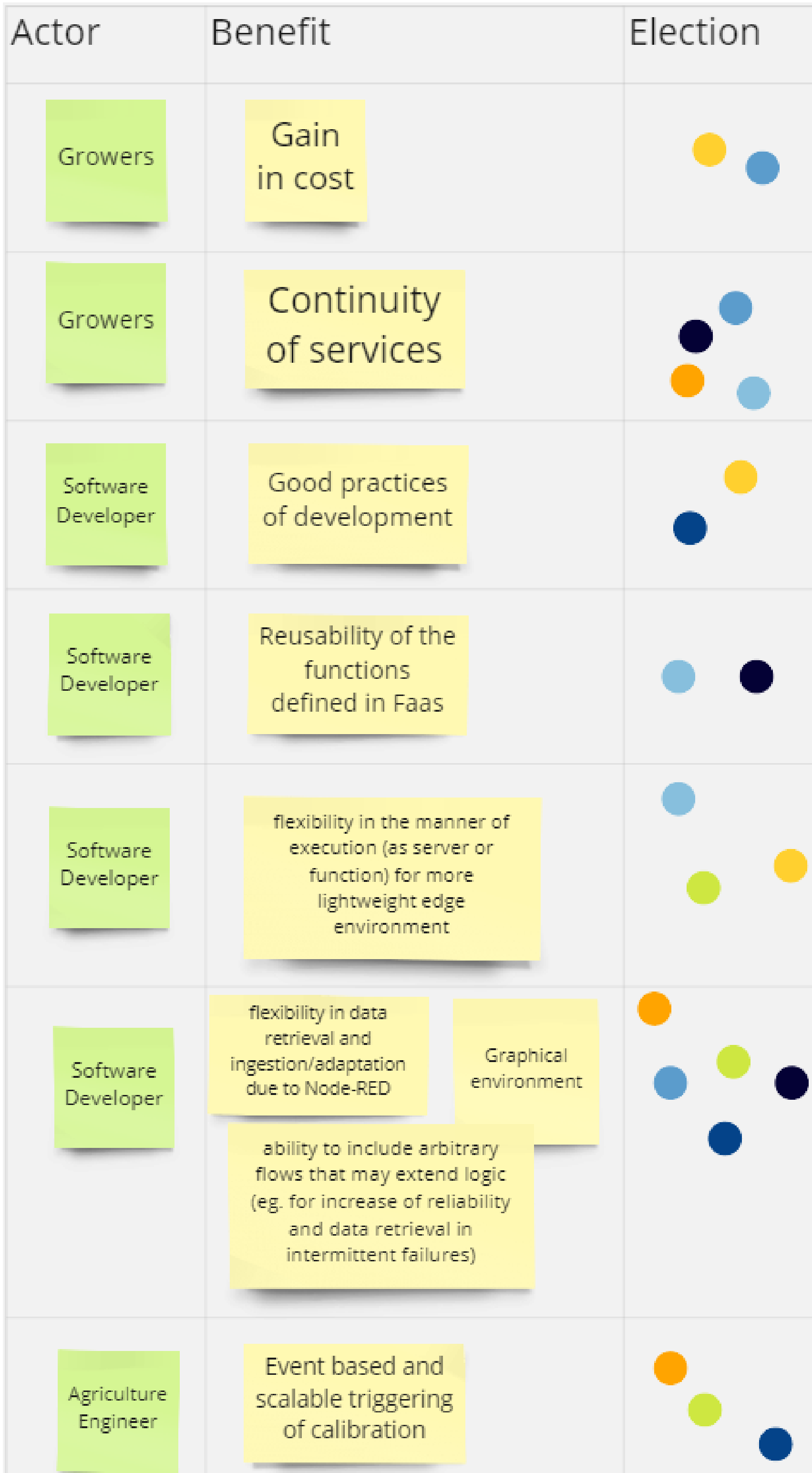

Figure 40 - Actor benefit prioritization of the Smart Agriculture Pilot

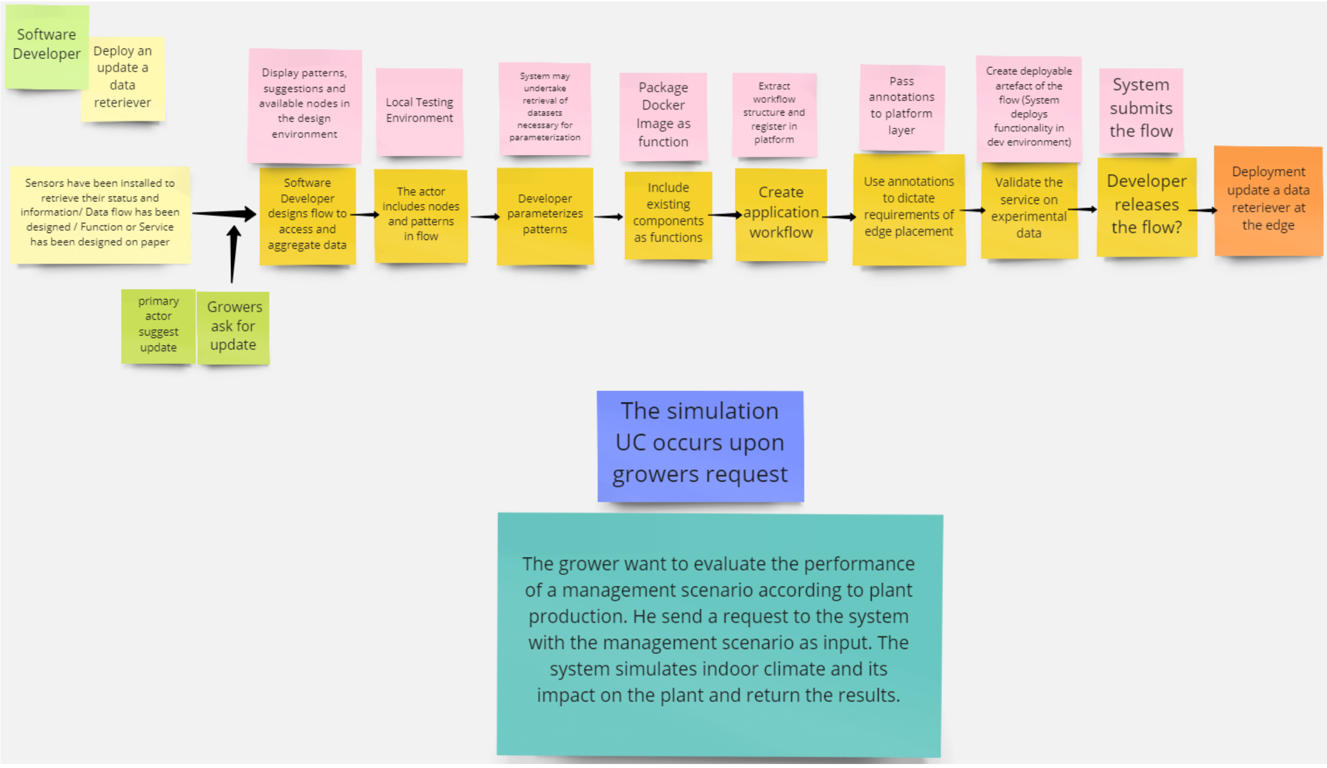

Figure 41 - Development of the Smart Agriculture Use Case #3.1

5.2 KPI Design Appendix

Figure 42 shows the second step of the KPI-Design workshop, which was used to identify the purpose of the KPI that is going to be developed and the prioritization among several goals.

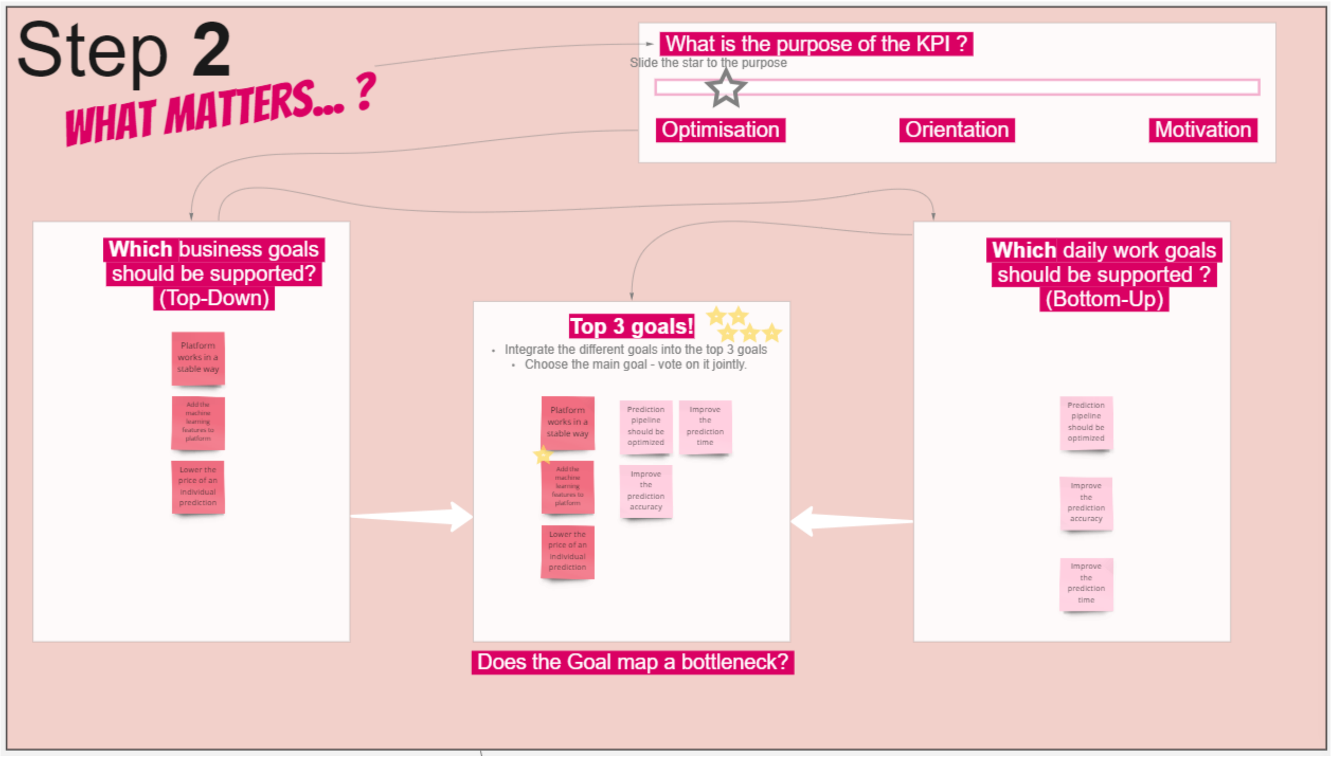

Figure 42 - Detail screenshot of the goal prioritization of the eHealth Pilot

Platform works in a stable way

Test A: Is the original goal a result? *If no, write down the most important intended result(s)*
NO
Mobile/ Webportal Application Requests are handled in 99% of the time within 1 second.
Test B: Are there any weasel words? *If yes, replace the weasel words with plain language and write the new result(s)*
NO
Test C: Is the goal multi-focus?
YES
Mobile Application Requests are handled in 99% of the time within 1 second.
Webportal Application Requests are handled in 99% of the time within 1 second.
Write your measurable performance result(s) here
Performance Result: Mobile Application Requests are handled in 99% of the time within 1 second.
Performance Result: Webportal Application Requests are handled in 99% of the time within 1 second.

Figure 43 - Detail screenshot of the quality check of initial formulated goals and the development into measurable performance results

DISCLAIMER

COPYRIGHT MESSAGE