Industrielle Gemeinschaftsforschung

Schlussbericht vom 22.12.2023

zu IGF-Vorhaben Nr. 303 EN

Thema

BlueSAM - Blueprints for Smart Products Architecture Management

Berichtszeitraum

01.05.2021 - 31.07.2023

Forschungsvereinigung

Forschungsinstitut für Rationalisierung FIR e. V. an der RWTH Aachen

Forschungseinrichtung(en)

Forschungsinstitut für Rationalisierung FIR e. V. an der RWTH Aachen

Sirris

Forschungsnetzwerk Mittelstand

Gefördert durch:

Bundesministerium für Wirtschaft und Klimaschutz

aufgrund eines Beschlusses des Deutschen Bundestages

The IGF project 303 EN of the research association FIR e. V. at RWTH Aachen University (Research Institute for Rationalization), Campus-Boulevard 55, 52074 Aachen, was funded by the AiF within the framework of the program for the promotion of joint industrial research (IGF) program by the Federal Ministry for Economic Affairs and Energy on the basis of a decision by the German Bundestag.

Introduction / Abstract

The transition to smart, connected products is in full progress and is one of the most important future trends and at the same time a growth market. For both Wallonia and Germany, this transformation is a necessity for a growing group of companies to remain internationally competitive.

Challenge

Designing smart products is a major challenge for SMEs in the tool, machine, and equipment manufacturing industry. Smart products consist of physical, software, and digital service components. SMEs often lack the competencies to design connected products and build larger software and digital service components, as their core business is to build the mechanical and electrical components of these products. Furthermore, their limited experience with smart products prevents them from accurately estimating and managing development costs and timelines, as well as operational costs. To be able to deliver working solutions, they need to co-develop the solution together with new partners using different processes and technologies, which involve a lot of risk. The state of the art does not adequately address these challenges are not sufficiently addressed by the state of the art.

Goal and Main Outcomes

Therefore, BlueSAM aimed to develop a method and guideline for the (co-)development of smart products from both a technical and a business perspective. This was done through architecture blueprints that provide architectures based on practical use cases. The blueprints were based on reference architecture and then further detailed in different views such as functional, technology, and business. The results of the project are presented via demonstrators and a web tool, ensuring maximum information distribution to SMEs.

Solution Path

At the beginning of the project, smart product use-cases were developed by case study research with members of the UC. At the same time, existing practical and standards-based reference architectures were examined for a smart product application in SMEs. A combination of two investigated architectures was selected as an architectural framework. Based on the framework and the developed use cases the BlueSAM reference architecture was derived. For each use case, specific architecture blueprints were designed and described in the architectural views. Afterwards, a method for the application of the blueprints and a company-specific customization was developed. The method is accompanied by a guideline that enables the co-development of smart products, with the blueprints as a general means of project management. Finally, the project results are presented in two demonstrators and in the BlueSAM web-tool, a free online tool was created to disseminate the project contents.

Benefit

BlueSAM enables SMEs to conceptualize, (co-)develop, and realize smart products. The individually developed blueprints help bridge their knowledge gap to smart products. From a selection of practical smart product use-cases a hands-on set of blueprints, and the needed method and a co-development guideline for turning their use-cases into business-cases.

BlueSAM is key for SMEs to successfully enter the smart products market, which is growing at 20-23 % annually.

This report was written by Sebastian Kremer (FIR) and Max-Ferdinand Stroh (FIR).

Baseline and Objective

This research project targets SME manufacturers of machines, equipment, and tools for different industrial sectors and technology partners that complement their value chain. A state-of-the-art architecture that ensures a technology platform for a successful transition from mechatronics to a new generation of smart, connected products is very important to these SME manufacturers [1]. Smart Products consist of a physical and a digital product and provide benefits to both, customer and manufacturer [2,3].

This chapter briefly outlines the baseline, the objective of the research project, and the state of the art at the start of the project.

Baseline

Driven by IoT and Industry 4.0, the evolution of products into smart, connected devices is radically reshaping companies and competition. The new product capabilities, the infrastructure, and the data they generate are reshaping core processes at product manufacturers, i.e. product development, IT, manufacturing, logistics, marketing, sales, and after-sale service, all with an increased importance of cybersecurity.

SMEs need to be able to enhance their product portfolio with software and digital service components to meet broader customers' needs and/or to adapt to changes in the value chains. However, designing smart products is a major challenge for SMEs in the tool, machine, and equipment manufacturing industry. Smart, connected products consist of physical, software, connectivity and digital service components. Therefore, they require SMEs to build and support an entirely new technology infrastructure. SMEs are typically highly specialized in their domain, but often lack competencies in these new domains to conceptualize an architecture and build digital service components. In addition, their limited experience prevents them from accurately estimating and managing costs and schedules or initiating co-development [4].

SMEs need architecture **tailored to their current and potential needs** from smart products. This architecture should enable them to develop their smart product, guarantee to capture a significant part of the value created by the new products and services without blocking possible future extensions. However, most of the existing smart product architectures are either theoretical and complex or made for specialized IT-professionals and beyond the capabilities of what an SME can implement (see State of the Art). There is a need for an accessible **method, tailored to SMEs, to "select" and "fine-tune" an architecture** for the technology platform of their smart product. It should define the main digital components required within the product, as well as the main components for the digital platform, enabling digital business models and services, which increase the value of the generated data.

Objective of the Project

The main objective of the BlueSAM research project was the development of assistance to SMEs that overall aims to lower the entry hurdle for SME to produce smart products. The focus was on the underlying IT-architecture that is required to realize and sustain the deployment of smart products.

This was set to be achieved by providing the SMEs with use-case-based, practical architecture blueprints and an appropriate method. The **architecture blueprint** represents a predefined architecture consisting of the main elements and modules to realize a use case.

Not all SMEs are able to timely develop or integrate all required skills. Building partnerships with digital service and software developers is often a more efficient and future-proof strategy. However, with limited experience in developing smart products, they find it **difficult to initiate successful co-development**. Regardless of the strategy chosen, SMEs encounter difficulties in maintaining control of their development, in terms of time, budget, and scope. Consequently, BlueSAM also aimed to address these challenges by developing a co-development guideline, which assists an SME to identify and approach partnerships for an accelerated development of smart products.

BlueSAM aims to bridge the gap in knowledge between emerging and commercially available technologies and their application in smart products, by developing an architecture development method, easy-to-use blueprints, and a concept for collaboration with technology providers. The objective of the developed knowledge is to advance the state of the art on industry-relevant topics in which SMEs are expressing a strong interest.

State of the Art

Reference architectures as basis for an SME platform

The IoT and Industry 4.0 vision propels a rapid technological evolution and requires a wide range of technologies and standards. Several consortia, associations, and companies developed reference IoT architectures, standards, and technology platforms. They can be classified as:

- Standards-based, interoperable architecture frameworks (i.e. IIC IoT Reference Architecture [5] (IIRA), Reference Architecture Model 4.0 [6] (RAMI 4.0), ISO IoT Reference Architecture [7], Open Industry 4.0 Alliance collaboration platform [8]) and theoretical reference architectures (i.e. New Technology Stack by Porter & Heppelmann [3][Fehler! Textmarke nicht definiert.]). These enable IIoT system architects to design their own systems based on a common framework and service-oriented concepts but are theoretical and beyond the scope of an SME.
- More practical reference architectures, generic for diverse applications across a broad spectrum of industrial verticals as well as specific to a particular field of application. Most are proprietary (i.e. Azure IoT Reference Architecture [9], AWS Well-Architected framework [10], ADAMOS-Platform [11], PTC ThingWorx, Bosch IoT Suite or Siemens MindSphere). Some are based on open source (i.e. FIWARE [12]). Their building blocks and tools offer a more pragmatic solution to develop smart, connected products. However, today's platforms have often been developed in the context of large, forward-looking companies. They offer far more functionality than an SME needs and require a large investment (or costly licensing model). Many commercial IoT platforms are linked directly to the offerings of large technology providers or machine/equipment builders and involve a risk of vendor lock-in.

As for their application by SME manufacturers who want to have their own smart, connectivity solution, these frameworks and platforms are high-level, too abstract and lack practicability. They include several conceptual layers necessary for smart products and are rather made for experts with profound IoT and software development knowledge (i.e. physical system, connectivity, edge application, cloud platform, services and applications, back-end information systems, communication to 3rd organizations). As a result, it is complex and time-consuming for SMEs to define their own architecture and to compare and select the technology platform and tools to initiate their smart product applications based on the state of the art.

SME-oriented methodologies to define a tailored architecture Providing architectures templates (combination of architecture modules) that are defined as “blueprints” in BlueSAM represents a significant improvement in the process of developing smart products The use of reference models assists SMEs by providing them with the necessary knowledge and increasing efficiency through the reproducibility of the results [13].

However, developing an architecture for an application is a challenging task. Few research has been performed so far to overcome this challenge. In their research, Krcmar et al. have developed a methodology for the development of reference architectures based on use cases and their requirements. This methodology was applied in the present study. [14].

Co-Development and Skills

A manufacturer of smart products needs to become a crossover between a software company and a traditional product company. Research highlights this additional set of skills needed to design, sell, and service smart, connected products [15]. These are in high demand but in limited supply and not easy to acquire by an SME. Different researchers have identified this skills shortage and have proposed guidelines for policy responses to reduce the skills mismatch (e.g. by education and training). Nevertheless, research lacks a method that enables SME-manufacturers to identify their skill gap and to close it in a timely and cost-effective manner, e.g. by being capable of initiating co-development with technology providers.

Project Plan and Achieved Results

Project Overview

The workplan structure was derived from the pursued research methodology, which was based on the case study research approach by Eisenhardt [16,17] and well-established methodology from Krcmar et al. based on the general processes of reference modelling [18,14,19]. The plan intended close collaboration with the user committee to steer developments towards the needs and absorption capacity of the target group of SMEs. The work packages (WP) were structured as follows:

- **WP1 Use-cases for Smart Products** collects and analyzes the use-cases pursued with smart products to derive generic use-cases to drive the research within the project.
- **WP2 State-of-the-Art and State-of-the-Practice Reference Architectures** analyses in parallel to WP1 the different reference architectures for smart products (interoperable frameworks as well as proprietary, as described in the state of the art).
- **WP3 Architecture Blueprints and Business Perspective** combines the results from WP 1 and 2 into the derivation of the BlueSAM architecture and the use-case specific architecture Blueprints. Those Blueprints are described in different views including the necessary cost and business components for their realization.

- **WP4 Methodology for Individual Architecture Application** defines an individual architecture blueprint and its views tailored to the use-case of an SME based on the modular BlueSAM reference architecture (WP3).
- **WP5 Co-Development Guideline** initiates and supports co-development with technology stakeholders by leveraging the methodology. The guideline will contain methods and processes for selecting and delegating tasks to suppliers.
- **WP6 Demonstration and Validation** applies and covers all aspects of the project on use-cases for new smart products as well as product enhancements (the architecture, its blueprints, the methodology, and the co-development guideline). Feedback will be integrated in WPs 3, 4, and 5. The web platform assisting the demonstration and validation will be developed.

The project was completed in accordance with the time plan as outlined in Figure 1. Yet, due to delivery delays for demonstrator parts, the project had to be extended (cost-neutrally) for two months. The project concluded with the final user committee meeting and smart product workshop on June 29, 2023, after a period of 26 months.

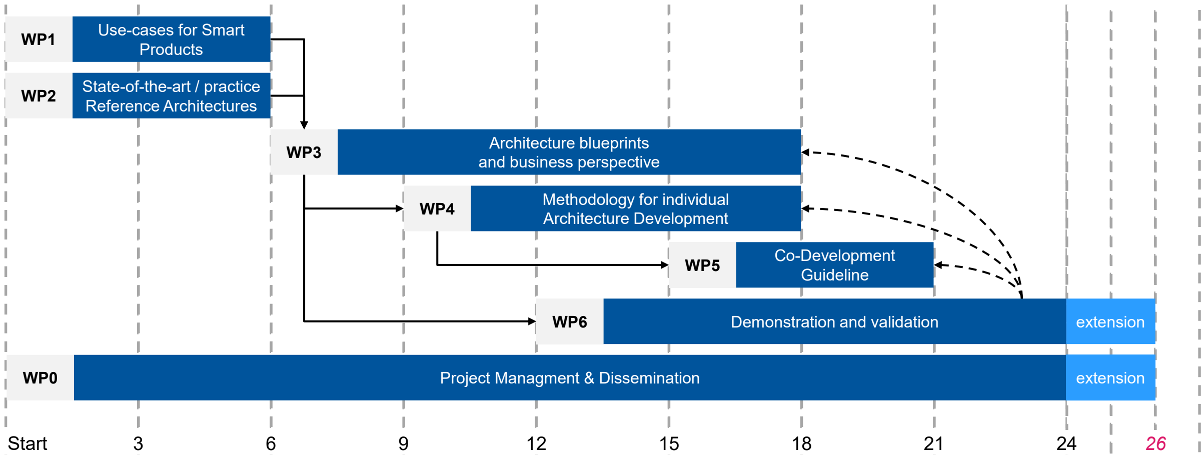

Figure 1 *Project plan and execution*

The following sections provide a detailed account of the objectives and execution of the work packages. The dissemination activities included in WP 0 are outlined in the chapter “Publications and Transfer Measures”. Therefore, they will not be discussed in further detail in the context of the work packages.

WP1 Use Cases (Lead FIR)

The objective of the WP 1 was to establish the architectural and methodological developments on the basis of relevant, practical use cases that apply to the project’s user committee. The aim was to focus the investigation and developments on tangible benefits for SMEs and to address the relevance of such use cases for the user committee at hand. To achieve this goal, the project team conducted interviews with the members of the committee about the practical application of (potential) smart products within their individual fields. Based on these interviews, a consolidated list of use cases was to be derived, which would then substantiate the base for the following work packages.

In the interview phase, the project partners FIR and Sirris conducted interview with the German and Belgian side of the committee and other participating companies, respectively. An overview of the interviews conducted is provided in Table 1.

Table 1 *List of Use Case Interviews*

Interviewee	Interviewer
grandcentrix GmbH	FIR
i2solutions GmbH	FIR
Sick AG	FIR
Janz Tec AG	FIR
Zentis GmbH & Co. KG	FIR
Oculavis GmbH	FIR
Mindsphere World e.V. (external)	FIR
BBK SA	Sirris
Belrobotics (external)	Sirris
Mircromega Dynamics SA	Sirris
Travelec	Sirris
Ecosteryl (external)	Sirris
Electronics Engineering – Consult & Design	Sirris
Eloy water	Sirris
Inductotherm Coating Equipment SA	Sirris
Iotree (external)	Sirris
PEPITe SA	Sirris
Stûv SA	Sirris
Sulzer	Sirris
Sweet move (external)	Sirris

The interviews conducted were analyzed and translated into a long list of use cases. By evaluating the list against a conducted literature review, the use cases were clustered into groups and consolidated into a list of generic use cases. The results were published in a conference paper in 2022 [20]. This list contains the following elements:

- **Analyze Usage Behavior**: This use case focuses on understanding the user's behavior by means of data analytics. It builds the foundation for a better understanding of how and why customers are using smart products. It means collecting the usage data such as frequency, duration, and location of usage as well as the used features from the smart product and storing it within the product cloud. In addition to that, user feedback on the product itself or certain features is added to analysis. By analyzing the usage behavior of a certain period of time, a behavioral profile of the user can be derived. The collected data is then matched with customer specific data as well as data about the customer's segment. In addition to that the customer's usage profiles can be compared to one another determining certain usage patterns.
- **Assist Operation**: Assisting the operation refers to helping the smart product's user to operate it. This means the smart product automatically guides its user. It monitors the way it is used and provides helpful information such as manuals, warning about dangerous or unintended misuse of the product, and prevents the user from making mistakes. It will allow remote assistance and remote control, as well as collect feedback from the users via its digital interface.
- **Condition Monitoring**: The use case condition monitoring consists of visualizing a products condition as well as providing the user with alerts when pre-defined thresholds are exceeded, or a set of rules triggers it. This means that the user is allowed to define alerts and rules. To execute the rules, the smart product collects data from different sources and aggregates them. The data may be from the product's environment, such as light, humidity (environmental data), from within the product, such as internal sensor

values or errors (product data), or from the process the product is in (process data). The later visualized or monitored values may be specific values, such as the vibration with a motor, or KPIs that aggregate different data streams. This could be the asset health or, for example, the OEE of the smart product. The condition monitoring use case serves as a base for many of the identified use cases. This use case can also be applied to the smart product's manufacturer monitoring certain KPIs for the smart product during its usage.

- **Create Digital Product Image**: A digital product image, product passport, or digital shadow provides an overview of historical data of the smart product. It stores all the historical product data and allows the user to explore it from different views. Such views may be the manufacturing history or the maintenance history of the smart product. Additionally, the data can be exported to other systems or be analyzed in a different context. A digital product image is very helpful for both the manufacturer and the user of the smart product.
- **Deliver Consumables/Supplies**: In this case, the smart product monitors the usage of supplies or consumables during its use. Based on the usage, it can estimate the remaining time until new supplies are needed. Being connected to the stock management system at the company or the supplier it can order material just in time to not run out of stock. This may also lead to a more detailed usage understanding for the product's manufacturer, also offering an additional value stream in case the supplies are directly sold by him.
- **Derive New Products/Services**: Next to understanding the user's behavior the collected data can be used to identify opportunities for new products and services. This means identifying completely new product ideas, missing product functions in existing smart products, or changing the configuration of features in a smart product. This means omitting unused functions or combining new functions. Thereby, the existing product configuration is measured against the identified customer needs from the usage behavior analysis and additionally identified needs from market research.
- **Improve Products/Services**: Improving the existing products and services based on usage data is another use case which was identified. It is very close to the derivation of new products. In this case the focus of the analysis is set on the existing functions and the way they are used. In addition to the usage data further information about the usage is collected from all customer interactions such as sales and service processes.
- **Offer Data Analytics**: This use case allows the smart product's user to individually deploy data analytics models on the smart product using its field and product data. This means selecting from a set of predefined analytics models that can be applied to the smart product and its environment. This can be anomaly detection, assistance for teaching sensor values, vibration, and temperature analytics and many more. It includes external data sources such as business systems, as well as storing and exporting the analytics results. Offering such functions may have a positive effect on the perceived value of the smart product.
- **Offer Subscription**: Offering a subscription business model means delivering continuous value improvement for a fixed fee. The online streaming service Netflix - with a fixed price rate and ever-growing offering - is an example of a subscription-based business model from a B2C context. The delivery of value is often measured in product performance. The improvement is achieved by analyzing the usage behavior to identify opportunities for process or product improvements for the specific customer use case.
- **Optimize Operation**: Optimize operation refers to helping the smart product's user to operate it at an optimal state. This is done via recommendations for the user to improve the product's performance or lifetime. The recommendations can be drawn from a predefined set of recommendations. The recommendations are selected on the basis of

an optimization model, constantly comparing the current operation parameters to an optimal state. The smart product's data can be enriched with external data sources like the production schedule from an MES or ERP software.

- **Provide Predictive Maintenance**: Predictive Maintenance aims at providing the user of the smart product with an interruption free usage period. It means identifying the need for maintenance and scheduling it before the smart products break down unexpectedly, leading to unforeseen downtime. In the case of an identified maintenance need, planned maintenance will be scheduled automatically to ensure maximum performance of the smart product. To do so, the remaining usable life of components or the whole smart product is calculated and monitored. This is done by connecting the as is data from the smart product with historical data and historical maintenance cases. Based on the identified issues, maintenance measures are selected, and maintenance is scheduled. As the component to be exchanged is known in advance, the maintenance personnel can bring the right equipment and plan the maintenance process accordingly. If needed, an exchange product can be provided during the maintenance period.
- **Update Products**: The Update product or service use case means improving the existing smart product by exchanging physical or virtual components of a smart product. To do so, an update needs to be ready for deployment. This may be a software bug fix but may also include the exchange of a physical component. Before that, the update needs to be authorized by the user. If necessary, maintenance is scheduled for the smart product. Depending on service level agreements, an exchange product may be delivered on site during the smart products update time.
- **Upgrade Products**: The process of upgrading a smart product is analogous to the update process. Meaning an upgrade is ready for deployment and will be executed similarly to the update. To identify the need for an upgrade, the usage of the smart product is monitored against predefined performance KPIs. Once a threshold is exceeded, the product use is evaluated. This could mean using a smart product at its upper or lower performance limit, triggering an exchange against a smaller or bigger product to better fit the customer's usage behavior. The upgrade is not limited to certain components of a smart product but may also contain the whole product itself.

Extending this list, further use cases were identified but not considered for further investigation. This was done, as these use cases were identified to be too broad in scope and possibly highly dependent on external factors that would complicate the specification of an individual IT-architecture in the scope of this project.

- **Offer service to 3rd party**: Generate value for a party outside the producer-customer relationship through the creation of a service based on the smart product's data.
- **Build a user community**: Create a social platform for customers to discuss and share product related experiences and expertise for enhanced customer bonding.
- **Create a platform for information exchange**: Create an open digital platform for customers and eventually externals to digitally share data and parameters.

As part of the main deliverable for this work package, each use case was further detailed in a UML-diagram to capture the constituting functionalities and document the use cases' scope as understood within the project BlueSAM. Furthermore, this allowed to more precisely determine the functional requirements per use cases for the architectural development in WP 3. The UML-diagrams for each use case are listed in chapter UML-diagrams for smart product use cases in the Appendix from Figure 31 through Figure 43.

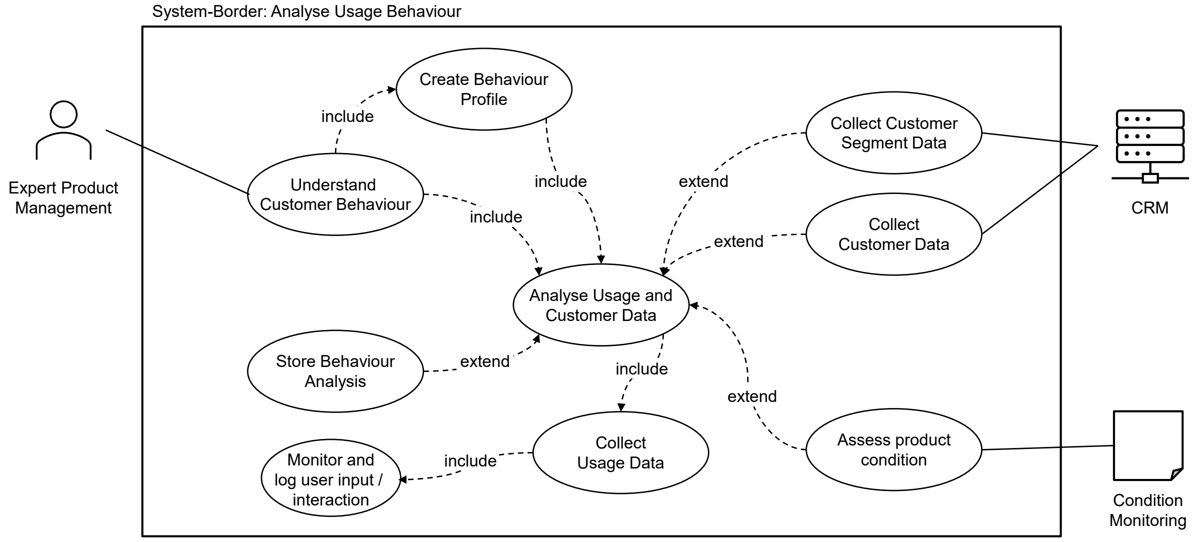

Figure 2 *UML-diagram for the use case "Analyze Usage Behavior"*

The UML diagram visualizes the relationship between stakeholders and technical systems based on functionalities that coordinate to fulfil the promises of the use case. The UML-diagram specifies these relationships by addressing what actions or functionalities require (“includes”) the presence of or extend on existing elements within the diagram. In the scope of this project, the relationship between the use cases was marked by including other use cases a stakeholder within the diagram. Hence, the UML-diagram for the use case “Analyze usage behavior” presented in Figure 2 states that for a product manager to understand the customers’ behavior, the functionality to create behavior profiles is required to reduce the complexity of the solution space. Additionally, the analysis of usage and customer data is supported by introducing collected customer data from a CRM system and assessing the product’s condition via the use case “Condition Monitoring”.

The developed UML-diagrams help to detail the underlying functionalities of the use cases for both the user studying the use cases and the development of architectural blueprints in WP 3. The definition of the business impact per use case was developed by utilizing Osterwalder’s Business Model Canvas. Because these did integrate into the final blueprints as a modular view, the creation of the business model canvases for each use case is elaborated in WP 3 and the resulting canvases presented in chapter Blueprints – Business Model Canvas in the appendix.

WP2 Reference Architectures (Lead Sirris)

The goal of work package two is to identify and define a reference architecture framework to be used as the basis for the development of the BlueSAM reference architecture. In the beginning, an overview of the existing reference architectures for smart products is created. Both practical and scientific architectures are taken into consideration. The architectures were analyzed based on the fit to be used as a reference architecture for smart products. Secondly, the collected architecture is reviewed together with the user committee. Finally, a reference architecture framework was defined based on the analyzed architectures and the feedback from the user committee. The research partner Sirris led this work package.

The following architectures were analyzed in the process:

- ISO IoT Reference Architecture
- Industrial Internet Reference Architecture
- Reference Architecture Model Industry 4.0 (RAMI4.0)
- Smart Product Technology Stack by Porter & Heppelman
- Azure Reference Architecture Framework
- PTC IoT Reference Architecture
- AWS Reference Architecture Framework
- Siemens Mindsphere Reference Architecture
- Google Cloud Reference Architecture
- ADAMOS Reference Architecture
- Alibaba Cloud Reference Architecture
- FIWARE Reference Architecture
- IBM Industry 4.0 Reference Architecture
- Bosch IoT Suite
- AVEVA (Wonderware) Reference Architecture

All of the above-mentioned architectures were researched and analyzed intensively. To ensure a scientifically accurate and valid analysis, the following criteria for analysis were chosen:

- **Expertise Requirements**: The level of specialized knowledge required to implement and manage the architecture effectively.
- **Modularity Aspects**: The ability to divide tasks among teams and ensure interoperability with other systems or platforms.
- **Availability of Different Views**: Whether the architecture offers multiple perspectives or dimensions to address the complexity of systems.
- **Smart Product Orientation / Specificities**: The focus on and support for smart products, including IoT devices and connected services.
- **Sensor to Cloud Coverage**: Coverage of the entire architecture from sensor level to cloud, including intermediate layers like edge and fog computing.
- **Security and Identity Coverage**: Emphasis on security measures and identity management across the system's components.
- **Device Management**: Capabilities for managing and maintaining device lifecycle, including updates and monitoring.
- **Digital Twin Coverage**: Support for digital twins, offering virtual representations of physical systems for simulation and analysis.
- **Data Governance**: Approaches to managing data integrity, privacy, access, and compliance.

- **Scalability**: The ability to scale the system to accommodate growth in data, devices, or complexity.

The detailed results were collected in a table and can be found in the appendix of this report. For better understanding for each of the selected criteria, a level of fulfillment from zero to 100 percent in 25 percent steps was calculated. The following table gives an overview of the calculation.

Table 2: Overview of analyzed architectures and the criteria fulfillment

Criteria / Architecture	Expertise Requirements	Modularity Aspects	Availability of Different Views	Smart Product Orientation	Sensor to Cloud Coverage	Security and Identity Coverage	Device Management	Digital Twin Coverage	Data Governance	Scalability
ISO IoT	100%	100%	75%	75%	100%	100%	75%	75%	75%	100%
IIRA	100%	100%	75%	75%	100%	100%	75%	75%	75%	100%
RAMI 4.0	100%	100%	100%	75%	100%	100%	75%	75%	75%	100%
Porter & Heppelmann	75%	50%	25%	100%	75%	75%	50%	50%	50%	75%
Azure	100%	100%	75%	50%	100%	100%	100%	100%	100%	100%
PTC	75%	75%	50%	100%	100%	75%	75%	75%	50%	100%
AWS	100%	100%	75%	50%	100%	100%	100%	100%	100%	100%
Siemens Mindsphere	100%	75%	75%	75%	100%	100%	75%	100%	75%	100%
Google Cloud	100%	100%	75%	50%	100%	100%	100%	100%	100%	100%
Adamos	75%	75%	50%	100%	100%	75%	75%	75%	50%	100%
Alibaba Cloud	100%	100%	75%	50%	100%	100%	100%	75%	75%	100%
FIWARE	75%	75%	50%	75%	100%	75%	75%	50%	75%	100%
IBM Industry 4.0	100%	75%	75%	75%	100%	100%	75%	75%	75%	100%
Bosch IoT	100%	100%	75%	75%	100%	100%	75%	75%	75%	100%
AVEVA (Wonderware)	100%	75%	75%	50%	100%	100%	75%	75%	75%	100%

The analyzed architectures were reviewed together with the user committee members during individual interviews and at the use committee meeting in January 2022. Also, individual architectures from the members of the user committee were taken into consideration. However, the selection of practical and scientific architectures proved sufficient. Finally, two architectures were selected as main components for the BlueSAM reference architecture. Instead of creating a new architecture as a reference, a practical and a scientific architecture were chosen.

For the scientific architecture, the Technology Stack from Porter & Heppelman was chosen for the practical architecture. The Microsoft Azure Reference Architecture was selected as a practical architecture.

The technology stack proposed by Michael E. Porter and James E. Heppelmann offers a comprehensive framework for understanding the convergence of technologies in the digital era, particularly within the context of smart, connected products. Their framework delineates the layers of technology required to create, connect, analyze, and integrate these advanced products into the broader ecosystem of Internet of Things (IoT).

The Porter-Heppelmann stack is structured around four primary layers, each representing a critical component in the development and deployment of smart, connected products:

Physical Product Layer: This base layer encompasses the actual physical product and its operational components, including mechanical and electrical parts. It is the tangible, real-world entity that users interact with.

Smart Components Layer: This layer adds intelligence to the physical product through embedded software, processors, sensors, and data storage capabilities. Smart components enable products to monitor, control, and optimize their own operations and usage, as well as communicate with other products and services.

Connectivity Layer: Connectivity components such as antennae, communication protocols, and networks enable the product to communicate data to and from the cloud, facilitating interaction with other devices, the manufacturer, and the environment.

Product Cloud Layer: This top layer, also known as the "product cloud", consists of the software applications, analytics, and services that support the product. It enables data collection, storage, analysis, and the execution of sophisticated functions like remote control, optimization, and autonomous coordination.

The advantage of scientific architecture is the clear distinction between the different components of a smart product system, and its direct focus on smart product use cases. The architecture can be seen in the following figure.

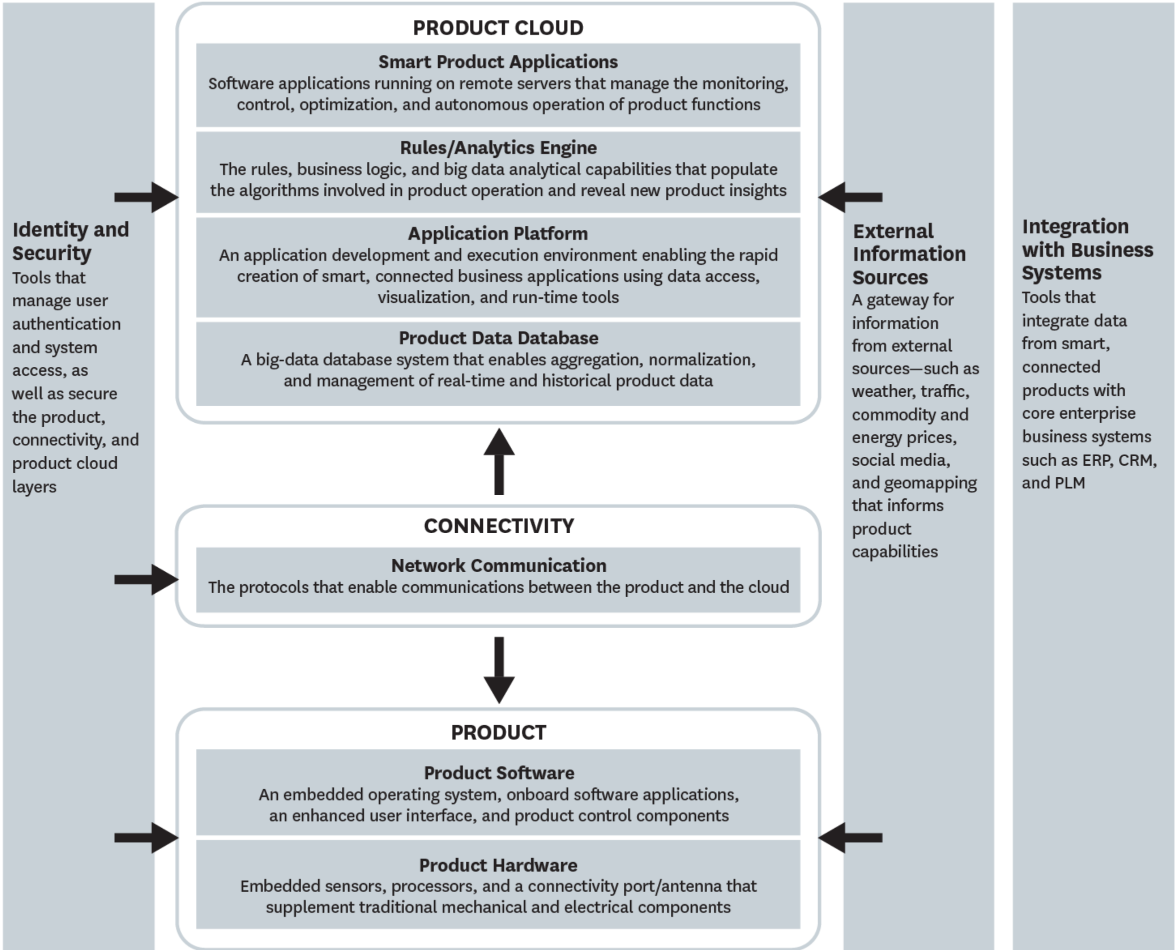

Figure 3: The "New Technology Stack" by Porter & Heppelmann

The advantage of the practical reference architecture, Microsoft Azure, is the distinction into functional components. For this reason, it serves as a reference for the function of you and the blueprints of the BlueSAM smart product architecture.

The Microsoft Azure IoT Reference Architecture provides a comprehensive blueprint for building scalable, reliable, and secure IoT solutions using Microsoft Azure services. This architecture is designed to guide developers and architects in the deployment of IoT systems, addressing common challenges such as device management, data analysis, and integration with existing infrastructure.

At its core, the Azure IoT Reference Architecture emphasizes a modular and flexible approach to IoT solution design, enabling customization and scalability. It is structured around several key components that work together to facilitate end-to-end IoT solutions:

Device Layer: This encompasses a wide range of IoT devices, including sensors, actuators, and embedded systems, responsible for collecting and transmitting data.

Gateway Layer: Gateways serve as intermediaries between devices and the cloud, providing local processing and storage capabilities, as well as secure connectivity to the cloud.

Cloud Gateway: Azure IoT Hub acts as the cloud gateway, enabling secure bidirectional communication between IoT devices and the Azure cloud platform. It supports device management and message routing functionalities.

Stream Processing: Azure Stream Analytics processes large streams of data in real-time, enabling complex event processing, data filtering, and aggregation.

Data Storage: Data is stored in scalable storage solutions such as Azure Blob Storage for unstructured data or Azure SQL Database for structured data, facilitating long-term data analysis.

Business Integration: Azure Logic Apps and Azure Functions allow for the integration of IoT data with enterprise systems, automating workflows and enabling decision-making processes.

User Interface: Solutions like Azure Time Series Insights and Power BI provide powerful visualization tools for data analytics, offering insights into IoT operations and performance.

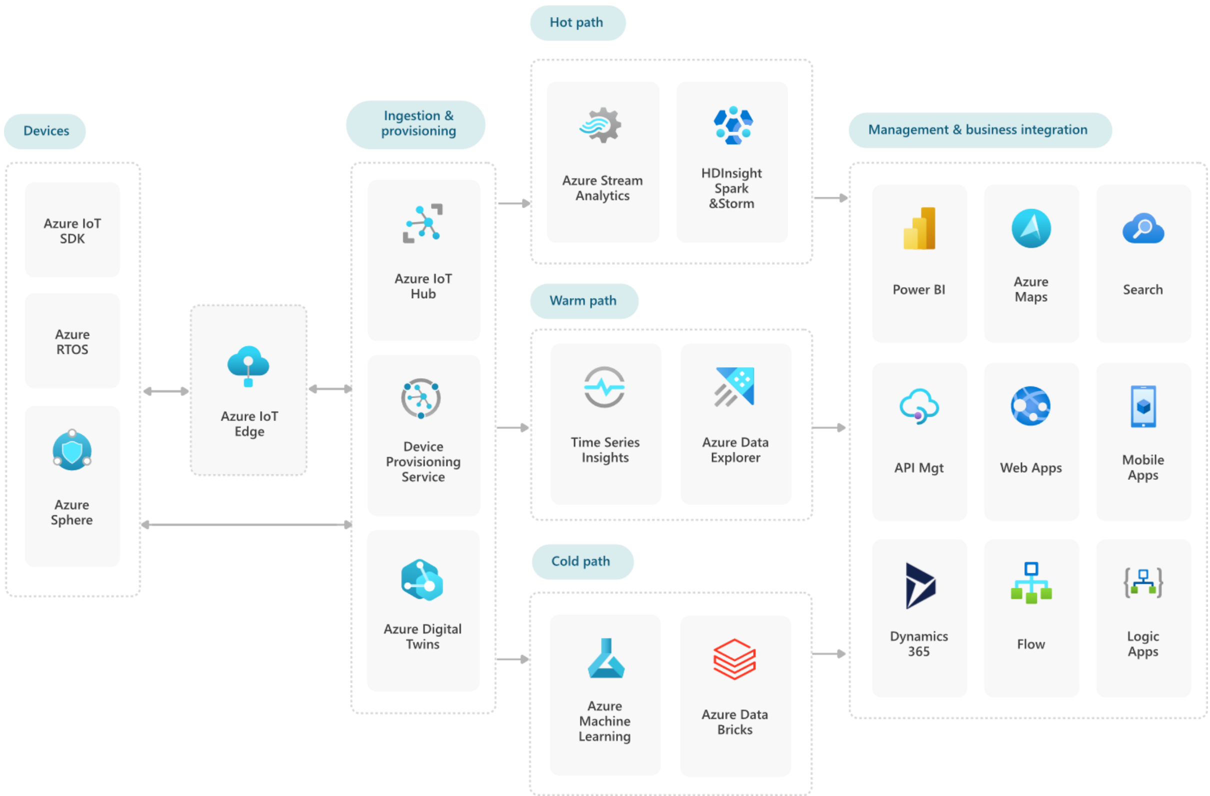

Figure 4: Azure IoT Reference Architecture

In addition to the two architectures that were selected as a framework, components from the RAMI 4.0 architecture were chosen to provide the basis for the different views for the BlueSAM reference architecture.

Conclusion: Different reference architectures were analyzed during WP 2. Based on the user committee's feedback and the analysis results, two architectures were selected as a base for the BlueSAM reference architecture. Instead of creating a new reference as a base, two of the existing architectures were selected for different views.

WP3 Architecture Blueprints (Lead FIR)

The objective of WP 3 was to develop use-case specific blueprints including a business perspective that allows the user to derive relevant components from the architecture. To achieve this, the task was set to first to derive requirements from the use cases, to generate a logical link from use case to blueprint. Based on these requirements, architectural components of the architecture defined in WP2 were set to be identified which then produced the base to construct the BlueSAM blueprints. The blueprints should be investigated for the expected business impacts for the implementing producer.

Derivation of requirements from use cases

The derivation of requirements was conducted on the long list of use cases and their respective UML-diagrams defined in WP1. First of all, the requirements were defined to represent **functional requirements** that communicate features or capabilities that the resulting system must include regardless of technical implementation or realistic systemic capabilities. This approach was decided as to describe what the perspective system is supposed to do instead of already limiting its architectural structures by defining how it would realize it.

Hence, the project team individually defined functional requirements for each use case to be consolidated into a final long list of about 300 requirements spanning across all use cases. These functional requirements define a short description, a corresponding use case, and a *necessity*. The defined necessity communicated its tie to the respective use cases in one of three qualities:

- "**must have**" (MH) rates a requirement that is essential to the operability of the use case
- "**should have**" (SH) rates a requirement that is expected or advised to be matched for a use case, but not functionally critical
- "**nice to have**" (NTH) rates a requirement that is purely optional for the operation of a use case and highly dependent on individual customization

As these functional requirements offer a direct tie to the use cases defined in WP1, they represent the "atomic" building blocks for the derivation of architectural components and method in WP4.

Definition of architectural components

Having defined functional requirements based on the use-case, the task was then set to create architectural components for the later construction of blueprints. Due to the focus on functionality and hence technically agnostic capabilities, the basic building block of the architecture was defined to be the **functional element**. The functional element describes an abstract capability of the architecture. They are substantiated by specific functional requirements of a use case and group into clusters.

In another iteration of consolidating the functional requirements defined for each use case, the functional elements were derived by identifying similar functionalities along the requirements. In this regard, a single requirement could constitute multiple functional elements, while a single functional element was mostly derived from multiple requirements. The derivation was discussed with the user committee and finally resulted in the definition of 27 different functional elements that represent the basic building blocks of the smart product architecture, as listed in Table 3.

The **cluster** of a functional element groups elements of similar concern and hence requirements of similar functional scope into a section that can possibly be identified in the architecture defined in WP2. The final identified clusters are:

- **API** groups requirements that provide functionality sets to establish and maintain inbound/outbound connection to an interface with a specific contract (REST/SOAP/RPC + documentation) to obtain or provide data (from and to our own cloud/edge infrastructure)
- **Data analysis** groups requirements that provide functionality sets to execute data science, from data exploration to automated data processing pipelines
- **Device management** groups requirements that provide functionality sets to maintain a list of devices for administrative purposes, with activation status, device-related information and to provision and update devices
- **Endpoint** groups requirements that provide generic interfaces sets to receive unstructured requests and provide information to the devices
- **Logic** groups requirements that provide functionality sets to execute logical operations to programmatically process data or realize/automate processes
- **Persistence** groups requirements that provide functionality sets to handle and store data over defined periods of time
- **Smart Product** groups requirements that provide a set of digital interfacing capabilities that the physical part of the smart product or its direct environment (sensors/edge) must offer to interact with the digital side
- **UI** groups requirements that provide user interface elements required to interact with the smart product system

The functional elements can be defined and clustered as follows.

Table 3 *Functional elements as basic building blocks of the BlueSAM architecture*

Functional element	Definition	Cluster
Connect to external API	Provides connectivity to a server to exchange information (mainly read) with external systems [API client]	API
Connect to business IT-System	Provides connectivity to a server to exchange information with business systems [API client]	API
Connect to internal platform	Provides connectivity to exchange information with other systems that are part of my internal infrastructure (e.g. database, broker) through an API [API client]	API
Provide API to serve RPC/data	This functional element exposes data or RPC via an API [API server]	API
ML model training	This functional element provides functionality to train machine-learning models	Data analysis
Data clustering	This functional element provides functionality to identify (manually or automatically) clusters of analysis results sharing some similarities	Data analysis
Data preprocessing	This functional element provides functionality to pre-process the data before analysis (e.g.: aggregation, feature computation …)	Data analysis

Data exploration	This functional element provides functionality for exploratory analysis of data	Data analysis
Data computation	This functional element provides functionality to apply a method, formula to a dataset or evaluate a model - model refers to any of explicit, statistical or ML	Data analysis
Remote update	This functional element provides functionality to update remotely the firmware/software deployed on the physical device.	Device management
Device metadata collection	This functional element provides functionality to collect metadata from the physical device itself or from the device management persistence.	Device management
Data ingestion	This functional element provides an endpoint to receive data from the physical devices or the gateways they connect to.	Endpoint
Data serving	This functional element provides an endpoint to serve data to the physical devices or the gateways they connect to.	Endpoint
Rule engine	This functional element provides a rule filter engine to trigger the execution of pre-defined functionality.	Logic
Scripting	This functional element provides an execution environment to execute a custom set of instructions.	Logic
Automation	This functional element provides the ability to execute or schedule the execution of a series of actions included in a process or workflow.	Logic
Database storage	This functional element provides data persistence in the form of database storage	Persistence
File storage	Provides data persistence in the form of file storage	Persistence
Capture relevant data	This functional element obtains raw data from sensors.	Smart Product
Receive data	This functional element receives data from the gateway/cloud/backend.	Smart Product
Transmit relevant data	This functional element sends data to the gateway/cloud/backend.	Smart Product
Actuate on command	This functional element receives a command to control a actuator and executes it.	Smart Product
Notification	This functional element provides proactive and intrusive alerting/messaging (not pull)	UI
Forms	This functional element provides an interface to input data (free text, checkboxes, basic form elements) in a predefined format (e.g.: form with a submit button)	UI
Low-code/No-code environment	This functional element provides an interface to define simple logic with drag-and-drop components and code snippets	UI

Control panel	This functional element provides interface to control the system live through forms technology with direct feedback, undo, … or through capability to input more complex data, extensive config options	UI
Dashboard	This functional element provides a visualization for KPI/charts/data with a limited & predefined interactivity (e.g.: selection in a dropdown list)	UI

Beside the functional elements listed above, more elements were identified from the grouping of functional requirements. These were, however, excluded from the construction of blueprints. The reasoning is that the further identified elements are either non-functional or address functionality that was deemed to concern all use cases when pursuing best practices and clean design strategies. As a result, these were defined as **transversal requirements** that need to be considered in any smart product development, but due to the generic nature are not relevant for the blueprint construction process. The identified and consolidated transversal requirements are:

- **Scalability** defines the ability to maintain operable and maintainable in a growing environment (necessity: MH)
- **Cybersecurity** defines the operative security against digital attacks (necessity: MH)
- **Observability** defines the ability to monitor and assess the state of the architectural components operational state (necessity: NTH)
- **Authentication/authorization** defines the ability to use-case specifically restrict access to networking, functionality, and data (necessity: SH)
- **Usability** defines the use of appropriate user experience for any stakeholder interacting with the smart product (necessity: NTH)
- **Speed** defines the ability to act and react in a timely manner (necessity: SH)
- **Robustness** defines overall operability in face of unknown/unforeseen conditions which relays to technical debt and is often handled by SLA (necessity: MH)
- **Mobility** defines flexibility regarding physical connectivity of the smart product and different deployment options for architectural components (necessity: NTH)
- **Confidentiality** defines the ability to act appropriately and securely with sensitive data in the context of data privacy and anonymization (necessity: SH)
- **Device management** defines the ability to manage product or customer related endpoints regarding provisioning, commissioning, fleet management, and identity management (necessity: MH)
- **Database optimization** defines the ability to handle technical debt in respect to data by employing cleaning, monitoring, and archiving mechanisms (necessity: SH)
- **Development process** defines the ability to develop, test and deploy by employing state-of-the-art development philosophies (e.g. agile and domain-driven-design) and toolchains (e.g. CI/CD) (necessity: SH)

The full list of functional requirements for each use case constituting a corresponding functional element is given in Table 10 in the Appendix.

Development of architectural blueprints

The development of the architectural blueprints was done in close collaboration with the development of the method in WP4. The later described intent to have the method be driven by use cases and to structure the underlying data in a graph-like manner, have largely determined the definition of the architectural blueprint as achieved in this project. Please refer to WP4 for a detailed explanation of why the derivation of blueprints from the functional elements was constructed in the way described in this chapter.

With the definition of functional elements and corresponding clusters as the central building blocks of the architecture, the project consortium discussed the different views that an architecture blueprint should provide in BlueSAM. While a functional view was a logical and undisputed first step, the other views were discussed in terms of intended communication and impact to the end user. The resulting blueprint views are **functional view**, **task view**, **data view**, **deployment view,** and a corresponding **business model canvas**. The description and intended goals of these views are presented below.

Functional View

The **functional view** of the blueprint consists of the previously defined main building blocks, the functional elements. This view presents capabilities that the smart product architecture has to offer, while specifying related functional requirements for each functional element that are defined by the underlying considered use cases. Based on the constituting requirements, the functional elements inherit a necessity qualifier defined by one of must have, should have, nice to have or disabled, in case no use-case-relevant functional requirement constitutes for the functional elements. The functional elements are further sorted by their cluster, to group related elements for a better overview.

The aim of this view is to communicate to the user the technology-agnostic capabilities that the intended architecture for operating the smart product has to offer. It confronts the user with the different aspects of realizing a smart product that both offer an overview of the problem at hand and highlights possible blind spots in development planning. Using this view, the user can identify capabilities that are already present in the existing infrastructure and more effectively address missing building blocks when discussing with experts or researching solutions on the market. In essence, the functional view provides terminology that allows for better communication and scoping of the problem / solution space.

Task View

The **task view** of the blueprint uses the existing functional elements and contextualizes them with a localized and processual concern. Each task element is derived from exactly one functional element and a single use case. The **task element** then defines the localization of the functionality by defining where in the architecture the capability might be located: the **domain**. The possible domains are:

- **Node**: a capability realized on the smart product itself
- **Edge**: a capability realized in the customers' infrastructure or area of influence
- **Backend**: a capability realized at the vendor's infrastructure or area of influence

Furthermore, the task element defines a processual **stage** during which capability might be needed to effectively meet the functional requirements. The processual stages are:

- **Data gathering**: the phase of collecting and introducing relevant data from endpoints or storages and transporting it to processing components
- **Data transformation**: the phase of transforming data with means of logical components and data analysis to derive use case relevant information.

- **Action**: the phase of executing measures derived from information gained during the data transformation process which can range from storing information for later use up to proactively executing actions regarding the smart product's use case.

Due to the generic nature of the blueprints, a task element can specify both multiple domains and multiple processual stages. The correct identification of where to realize a capability must be decided by the end user in the context of their specific use case.

The aim of the task view is to make the user aware of where and when an architectural capability must be present. This is to address blind spots that arise simply from misunderstandings about situation and processual stages that could lead to functionality being realized in the wrong context, making it either ineffective or even useless. Specifically, this intends to address the correct and logical construction of data pipelines. With the task view the user can effectively ask themself critical questions about the correct domain of application and logical causality of functions to prevent ill-conceived measures, for example whether the presence of a capability in an existing IT-system does fit into the smart product's use case or must be realized elsewhere to prevent complex restructuring.

Data view

The **data view** of the blueprint employs **data elements** derived from task elements and contextualizes them with locations. The data elements communicate the type of data a task element might be handling in its situation and processual stage. The data element is defined by an exemplary data point, which is contextualized into one type of data within the context of a concerned cluster and location of origin (=domain). The possible data types and clusters are listed in Table 4, Table 5, and Table 6 for the domains customer, public and vendor respectively:

Table 4 *Data types and cluster from the customer domain*

Data cluster	Cluster description	Data Type	Type examples
Customer	Customer data describes the customer to the vendor in the scope of their relationship regarding contractual obligations via the smart product.	Contract	Allowance, testing group
		Master Data	Address, billing
Local Environment	Local environment data describes the production/field environment of the customer including workers/users. It comprises intelligence (such sensor data) from the factory/field and feedback by users.	Field	Temperature, humidity, (day)time, load
		Feedback	Customer Feedback, marked issues
Process	Process data is produced during and directly linked to the usage of the smart product. These include parameter input, measurements	Input	Machine settings, parameters, physical input
		Output	Order data, Quality data

	during process, interaction with user and parameter output.	Live	Sensor data, user input/interaction, current operation
Device	Device data describes the smart product device to the vendor in the scope of processual history, factory and current configuration and its operational state.	History	Usage stats, maintenance history
		Configuration	Software/Firmware, Hardware
		State	Operational, Health

Table 5 *Data types and clusters from the public domain*

Data cluster	Cluster description	Data Type	Type examples
Global Environment	Global environment data consists of any data accessed by external sources through (often public) websites/API and interfaces to suppliers/sellers.	Supply chain	Supply integration, resource availability, discrete demand
		Field	Weather, traffic, events
		Market	User analysis, competitor analysis, resources, general demand

Table 6 *Data types and clusters from the vendor domain*

Data cluster	Cluster description	Data Type	Type examples
Configuration	Configuration data represent the flexible parameters to the data transformation process of the smart product's service.	Parameters	Payment metrics, update/upgrade plan, Low-Code configuration of Dashboard
		Transformation	Scripting, automation, data analysis pipeline
		Return	UI, rule engine
Asset	Asset data describes information that is a long-time resource to the smart product vendor. It includes resources for the product's use, gained knowledge by the vendor and lateral sources as other use cases.	Knowledge	Collected measures, known issues, know-how
		Lateral	Other use cases, available configurations, IT-system
		Product	Manuals, software/firmware, previous analyses, (longtime) collected data

Result	Result data represents any information that is gained during the data transformation process. It delivers the foundation for any notification, measure or learning proceeded from the transformation process.	Learning	New asset, derived strategical decision, development of product
		Information	Derived information, notification, alert, usage advice, ruling/decision-making
		Action	Measure taken, billing, remote control/configuration

The goal of the data view is to sensitize the user to the data that must be handled in the smart product's use case in the context of its origin and eventual data logistics. It serves both the identification of relevant data for the use case and the correct identification of data origins to address issues such as ownership, interfaces, formats, and transport. Even though the data view is not able to specify data points for the individual use case of the user, the clustering of generic data points aims to signal towards possible blind spots or misconceptions regarding the data's availability and accessibility. Finally, the structure is also intended to assist in the utilization of attained information in respect to the construction of data/information pipelines and feedback loops (where data/information to be fed back is produced at a later processual stage).

Deployment View

The **deployment view** is derived from the functional elements and provides an overview of technologies that offer capabilities described by the functional elements. This view is a result of extensive research on available products on the market and their functionalities regarding smart product use cases. The technologies identified are defined by a brief description and a classification of the type of technology, the rollout/deployment options and the estimated level of expertise required from the user. These classifications are structured as follows:

- Classification of technology types
    - **Protocol**: data exchange format to realize data logistics
    - **Service**: service that is publicly or privately accessible by integration into own logic or within a third-party ecosystem
    - **Software**: software that is distributed as a standalone package
    - **Product**: hardware product that is extending own infrastructure
- Classification of rollout options
    - **On-site**: deployable on own infrastructure
    - **IaaS**: deployable as infrastructure as a service
    - **PaaS**: utilization as platform as a service
    - **SaaS**: utilization as software as a service
- Classification of expertise levels
    - **User**: average user without special expertise
    - **Power user**: user with specific expertise within the professional domain
    - **Dev**: software developer utilizing planning, building, and testing expertise
    - **DevOps**: software developer additionally utilizing deployment, operation, and monitoring expertise
    - **DevSec:** software developer additionally utilizing modern IT-security practices
    - **DevSecOps**: software developer utilizing all of the above *dev*-tasks

The goal of the deployment view is to offer first leads towards available technologies that can be utilized to realize the functional elements of the functional view. While the generated list does not represent an exhaustive collection of available technologies, it does give first pointers as to what technologies do exist. Furthermore, the classification allows the user to identify those that fit the individual strategy of module deployment and match necessary expertise with existing and/or to-be established skills.

Business Model Canvas

Finally, the b**usiness model canvas** offers additional business context with the selected use cases. In contrast to the other views, the business model canvas does not derive from the functional elements but were constructed directly from the use cases defined in WP1. Hence, its structure is identical to Osterwalder's business model canvas from 2006 that defines the business model from the dimensions of value proposition, key partners, key activities, customer channels, customer relationship, customer segments cost structure, and revenue streams. Use-case-specific **business model elements** were defined in BlueSAM that belong to one of these dimensions and qualify as either an implication, so as an element that must be considered, or a potential, which is optional or represents an opportunity.

The goal of the business model canvas is to offer the user a business perspective in the context of the considered use cases. While other views lean more towards the technical architecture of the smart product, the business view contributes an organizational and business strategical dimension. Using the business model canvas implications and potential of different business dimensions can be individually assessed to construct the overall business case of the smart product.

Adapted during project: processual view

During the application phase of the research project BlueSAM, a processual view was mentioned and considered to be included in the blueprints. While developing this view produced **flow elements** that represent a process procedure for each use case, interviews with the project's user committee revealed that a generic view of the process is not always applicable. By incorporating a processual stage in the task view, the contemplated goal with the flow elements could be repurposed without adhering to a strictly defined procedure. Thus, the adapted processual view is fully compensated by the task view.

Final blueprints

With the definition of all views, the use case-specific blueprints could be constructed. As further explained in the developed method in WP4 the desired customizability of blueprints required for the blueprints to be generated from a baseline that is extended upon with individual constraints. The baseline itself is derived from the initially defined functional requirements which allows for the blueprints to be constructed from data tables. A graphical construction of these blueprints for visualization is elaborated in WP4 and publicly available online at bluesam-tool.fir.de.

The blueprints for each use case's functional view can be found in Table 10 in the Appendix, referencing the related functional requirements. It holds a collection of functional elements grouped by their respective clusters.

The blueprints for each use case's task view can be found in Table 11 in the Appendix, referencing the related functional requirements: It holds a collection of tasks which are represented by a functional element mapped in a domain (situation of where the capability might need to be established) and a (processual) stage.

The blueprints for each use case's data view can be found in Table 12 in the appendix, referencing the related task elements. It holds a collection of possibly relevant data points grouped by their respective data types and clusters elaborated above.

The blueprints for each use case's deployment view can be constructed from Table 13 in the appendix, referencing the related functional elements. It holds a collection of possibly relevant technologies rated for rollout options and required expertise level. The table does explicitly not provide a list of technologies for each single use case, as such a collection would contain over 4000 entries – instead the researched technologies are listed once, referencing functional elements that they enable. Applying the functional view to the deployment view, relevant technologies can then be identified.

The blueprints for each use case's business model canvas can be found in Table 14 in the appendix. It holds a collection of induced and potential business model elements that state business relevant factors grouped by their respective cluster from Osterwalder's Business Model Canvas.

Exemplary blueprint for use case Analyze Usage Behavior

As an example, on how to visualize the blueprint's views, the architectural blueprint for the use case analyzes usage behavior is shown below. The visualizations were generated using the publicly available webtool at bluesam-tool.fir.de.

The functional view visualizes the clustered functional elements and marks their relevance to the concerned use cases by its styling: greyed out elements are irrelevant for the use case, normally colored rectangular boxes define a "must have" functionality, dashed borders define either a "should have" functionality or when styled with lighter coloring a "nice to have" functionality (see Figure 5).

Figure 5 *Blueprint for Use Case Analyze Usage Behavior - Functional View*

The task view visualizes the tasks, derived from the functional elements, into a matrix of domains and (processual) stages, to highlight their applicability along these dimensions (see Figure 6). As described above, task elements can occupy multiple segments of the matrix as the specific placing is highly dependent on the individual case. In general, the use cases point towards a sideways U-shaped flow of functionality from data gathering at the node and data transformation at the backend towards derived actions back at the node or edge.

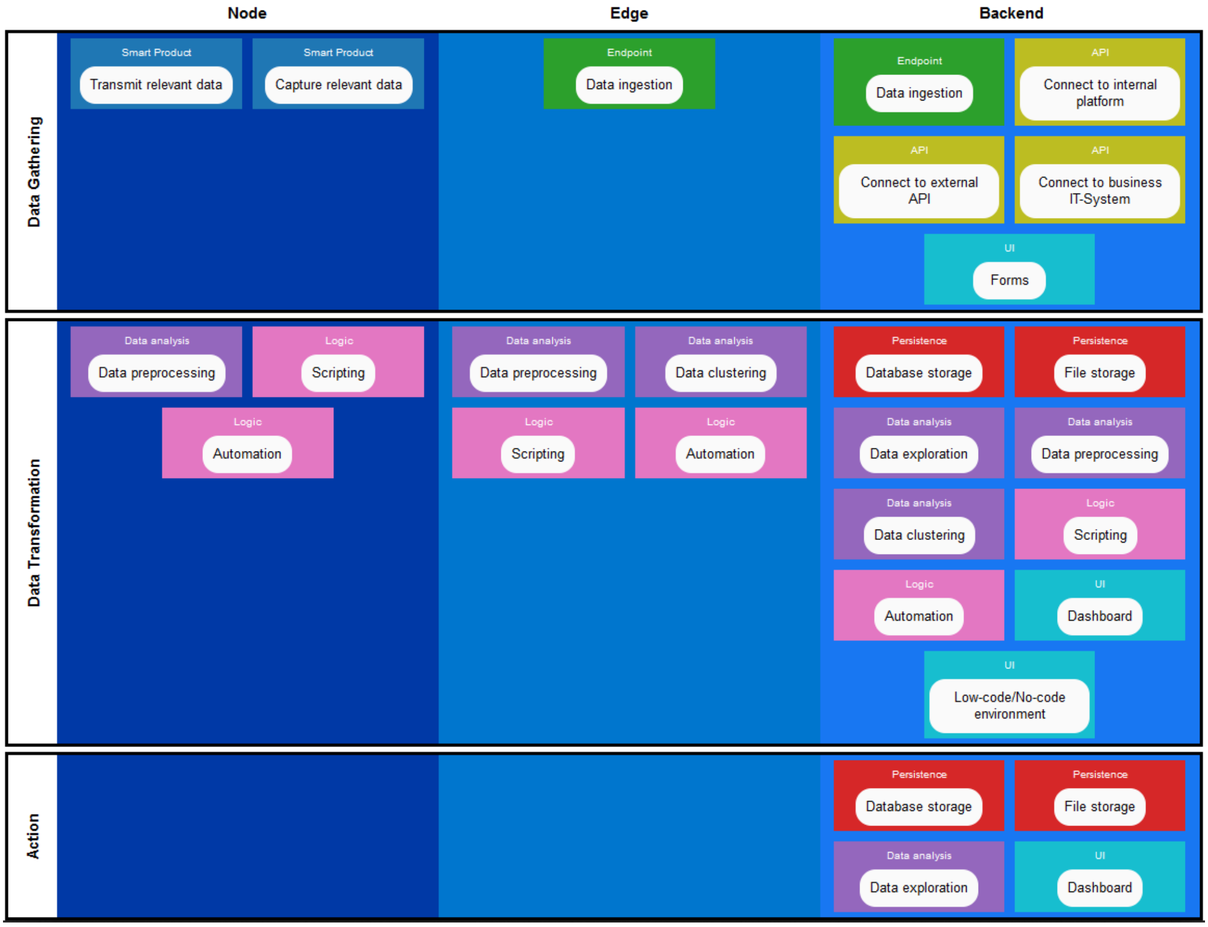

Figure 6 *Blueprint for Use Case Analyze Usage Behavior - Task View*

The data view visualizes the possibly relevant data points, handled by the task elements, into segments of different cluster, to highlight their origin. The clusters themselves are collected as described above into different domains: customer domain at the top left, public domain at the top right and vendor domain at the base (see Figure 7).

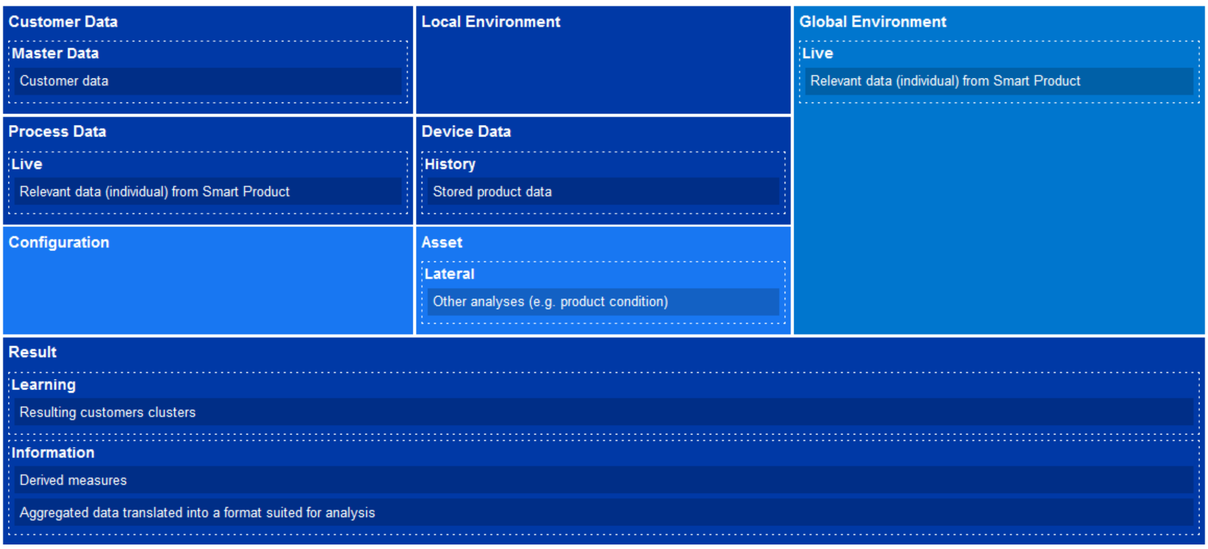

Figure 7 *Blueprint for Use Case Analyze Usage Behavior - Data View*

The deployment view lists the possibly relevant technologies that enable functional elements. Visualization limits itself to a long list of relevant technologies for each considered functional element. An excerpt of this list is shown in Figure 8. A full list can be derived from chapter

Blueprints – Deployment View in the Appendix.

UI - Dashboard

This functional element provides a visualization for KPI/charts/data with a limited & predefined interactivity (e.g.: selection in a dropdown list)

Title	Type	Description	Rollout options	Level
Amazon Quicksight	Service	BI tool	SaaS	Power user
AWS CloudFront	Service	Content Delivery Netwrok	SaaS	DevSecOps
AWS Cognito	Service	Authentication/authorization	PaaS	DevSec
AWS Elastic Beanstalk	Service	Web App delivery	PaaS	Dev
AWS IAM	Service	Authentication/authorization	SaaS	DevSec
AWS S3 (Simple Storage Service)	Service	File storage (storage or web front-end publishing)	PaaS	DevSecOps
Azure Active Directory	Service	Authentication/authorization	PaaS	DevSec
Azure App Services and Cloud Services	Service	Web App delivery	PaaS	DevSecOps
Azure Blob storage	Service	File storage (storage or web front-end publishing)	PaaS	DevSecOps
Azure Content Delivery Network	Service	Content Delivery Netwrok	SaaS	DevSecOps
Azure IoT Central	Software	IoT Platform	SaaS	Power user
Google App Engine	Service	Web App delivery	PaaS	DevSecOps
Google Cloud storage	Service	File storage (storage or web front-end publishing)	PaaS	DevSecOps
Grafana	Software	BI/visualization tool	On-Site, IaaS, PaaS	Power user

Figure 8 *Blueprint for Use Case Analyze Usage Behavior - Deployment view (excerpt)*

Finally, the business model canvas for the use case Analyze Usage Behavior is presented as per Osterwalder's Business Model Canvas. Business model elements that communicate a potential instead of an implication are formatted in italics (see Figure 9).

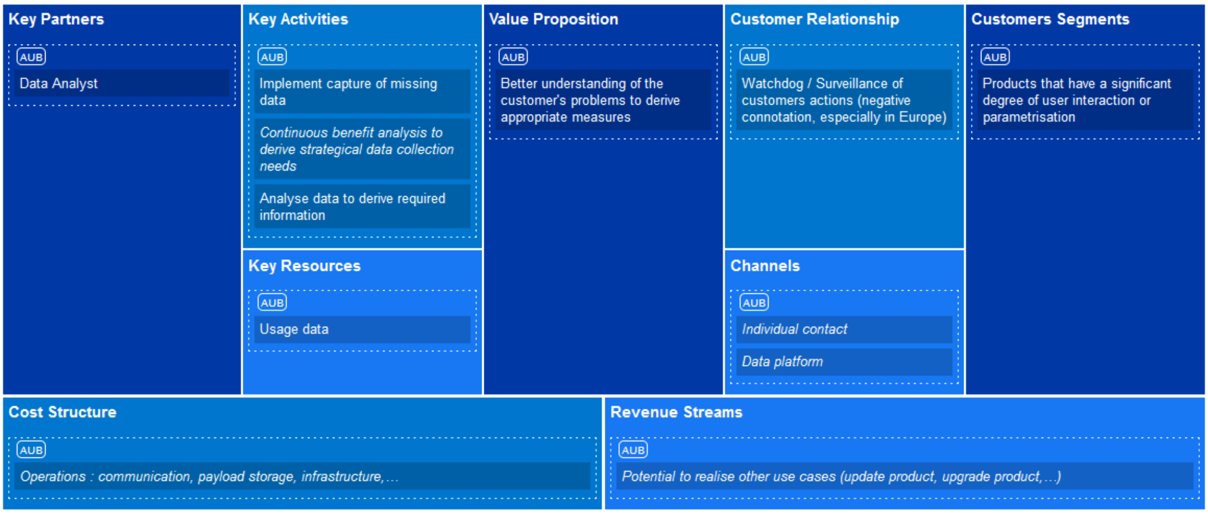

Figure 9 *Blueprint for Use Case Analyze Usage Behavior - Business Model Canvas*

As for every blueprint, the transversal requirements as described above must be taken into account in the development of a smart product. An additional information sheet for the transversal requirements is presented in chapter Additional Content for Blueprints – Transversal Requirements in the Appendix.

WP4 Methodology (Lead Sirris)

The objective of WP 4 was to develop a method that enables SME to individually assess and shape its architecture based on the blueprints from WP 3. The overall goal was again to lower the barrier for SME to start developing an individual smart product, based on use cases that directly communicate a perspective benefit. From the beginning of WP 4 and during the already started WP 3, it became clear to the project team that the development of the blueprints needed to be aligned with the development of the project’s deliverable. Otherwise, the benefit of the blueprints would suffer from suboptimal communication of the project results.

Developing the method

The work package started with a critical look at what requirements shape the blueprinting process. While the method should be use-case-oriented, as described above, SME should not be required to conduct a lengthy self-assessment process to yield first results. So, the method is required to generate default blueprints simply from the selection of one or more use cases. Another non-functional requirement that the project team identified for the method was that the availability of the blueprints should be enhanced by a publicly available web tool (as planned in WP 6), which implements the method for SME to use with digital assistance. This is turn required the method to at best employ existing mechanisms of a software framework, so that its process is easily implementable.

Looking at the preferred web development framework for WP 6, React.js / Next.js, an overall mechanism for the method could be defined. As this framework relies on a definite state to derive results (generated webpages) from, the process would need to incorporate trackable changes to that state which deterministically lead to a changed result. In translation for the method this meant, that the default state of the use case selection requires individual user changes to be built into the inner workings of the method, so that an individualized blueprint is not only always presentable to the user but also incrementally changeable.

This understanding led to a core principle of the method: the functional requirements that are intrinsically linked to the use cases represent the customizable building blocks of the method. Hence any blueprint creation and customization need to be realized via a change of these functional requirements. This way default blueprints are easily definable just by applying the identified requirements from WP 3 and the user can introduce individual changes by evaluating in the professional context of the use case whether these default requirements do rightly translate to the individual case. But simply disabling these requirements, a new, individualized blueprint can be derived.

The definition of the method framework now created another problem to solve, as to how to derive all blueprint views from just the functional requirements, logically and digitally. With the requirement of implementing the method within a web app, the technology of graph database was considered. Graph databases allow to define specific object and individual relationships between these objects. In contrast to classical relational databases, graph databases are not required to define restricting object tables and have no need to define relationship tables. All these data models are abstracted in an easily query-able language that immensely assists in the digital implementation of logical relationships.

Based on the graph database framework of Neo4J, the blueprints from WP 3 were designed to adhere to a relationship model that was parallelly implemented in such a graphical database for later implementation. The developed relationship model is defined in Figure 10.

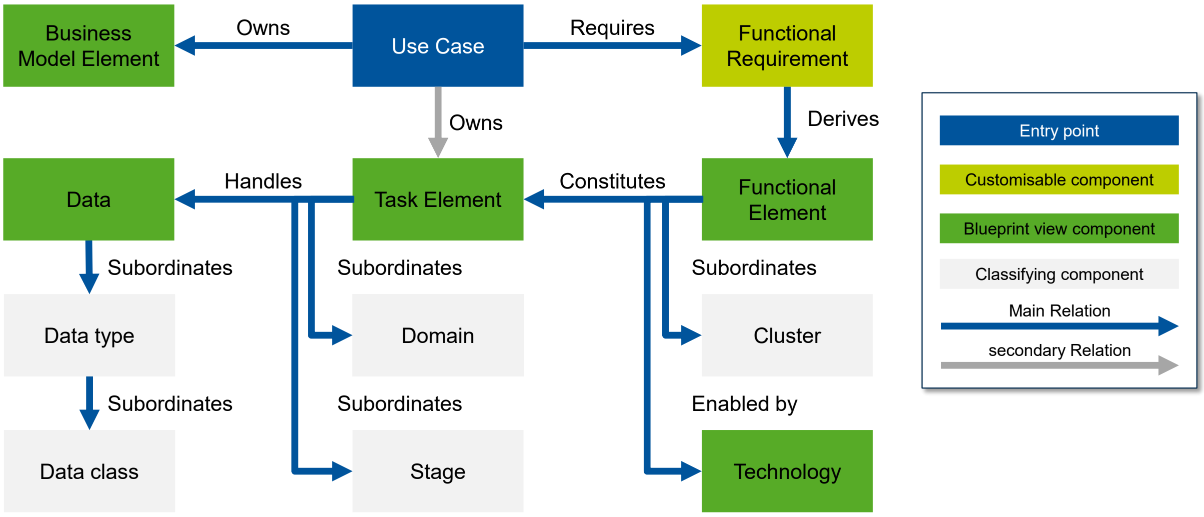

Figure 10 *Relationship model of the blueprint components*

Finally, the design of the blueprints from WP 3 were discussed in a way to deal with the selection of one or multiple use cases, so as that they consolidate each view into a combined blueprint for all selected use cases. This way, all possible customization paths can be handled by the methods relationship model.

Accompanying the method, additional materials were created to assist the architecture development process for the user. As the smart product architecture is to be expected to integrate into the general enterprise IT-architecture, as the implemented use cases directly concern the vendor side, the decision was made to orient these processes on the more general enterprise architecture management, EAM. Regarding the EAM the TOGAF method was identified to assist the user in the development and continuous management of the architecture. Additionally, the process of assessing organizational skill levels for the realization of a smart product, is addressed in the following work package WP5 Co-Development Guideline (Lead Sirris).

Developed Method for Selection and Customization of Blueprints

The developed method was mainly influenced by the underlying framework designated for its technical implementation into a publicly accessible webtool. Having defined the relationship model, it structures the path for both the method and implementation. The method is as follows:

1. **Selection of use cases**: The first step of the method requires the use to assess the desired use cases. The goal of this step is to prevent the use from utilizing a technically oriented approach and instead aim for a benefit-oriented development of a smart product. This addresses the often-overlooked dilemma of implementing solutions for non-existing or wrongly recognized problems and hence no creating a benefit. By first looking at the desired case in actual application of the product, the use is forced to assess whether the solution to be implemented is directed at a valuable goal.
2. **(Optional) Customization of functional requirements: Once the desired use cases have been**, the method can already default to generic blueprints (this is where the digitally implemented method excels, as any following steps are by design automatable and iteratively generate new results). Any in-method customization is to be done at this stage of the method, by filtering for functional requirements that are relevant to the individual case. By utilizing the contents of Table 10 in the Appendix, functional elements derived from the respective functional requirements (in the following step of the

method) can be omitted or rated with different necessity. To address different professional views on the matter, the respective clusters should be either for internal discussions on functional requirements or as a basis for consulting with experts.

3. **Derivation of functional elements:** With the utilization of Table 10 in the appendix, the selected use cases (and eventually curated functional requirements from step 2) produce a subset of functional elements, from which the functional view of the blueprint is to be constructed. By applying the necessity ratings for each functional element and grouping the elements by their clusters, a simple overview of relevant architectural capabilities is derived.
4. **Derivation of task elements:** Similarly, to the functional elements, the task elements are derived by utilizing Table 11 in the appendix, by filtering for the selected use cases (and eventually curated functional requirements from step 2). The identical naming of task and functional elements allows to further filter the tasks by eventually curated functional elements from step 2-3. By applying the domain and stages qualifiers for each task element, a matrix as described in chapter WP3 Architecture Blueprints (Lead FIR) is derived.
5. **Derivation of data elements:** Analogous, to the task elements, the data elements are derived by utilizing Table 12 in the appendix, by filtering for the selected use cases. Data elements can further be filtered by the eventually curated task elements from step 4. By applying the data type and cluster qualifiers for each task element, an illustration as described in chapter WP3 Architecture Blueprints (Lead FIR) is derived.
6. **Derivation of technologies:** Based on step 3 the technologies can be directly derived from the functional elements by utilizing Table 13 in the Appendix. Instead of filtering for the use case, the technologies are directly derived from the previously defined functional elements. In contrast to the other views, the technological view itself serves an exploratory character. To identify relevant technologies, the user must at least address respective rollout options and expertise levels. At best, it serves as a base to conduct further research with experts once a clearer picture of the architecture has been obtained.
7. **Study additional material to assess individual strategy:** Along with the blueprints the project yielded further information to be considered by the user in their development process. The transversal requirements, as per chapter WP3 Architecture Blueprints (Lead FIR), are to be considered when defining an underlying IT-Strategy for the realization of an architecture and can be applied as general design principles to be uphold. The continuous management of the developing architecture utilizing EAM and TOGAF is explained in the chapter Additional content for blueprints – Architecture Management in the Appendix and should be considered to establish an EAM process which in turn deals with the smart product architecture as part of the enterprise IT-architecture. Finally the consideration of organizational capabilities is to be assessed regarding the chapter Additional Content for Blueprints – Co-development Guideline in the appendix, which will be elaborated on in the following work package WP5 Co-Development Guideline (Lead Sirris).

Influence by the webtool's technologies

To assist the user in the application of the method and for publication purposes, the method was implemented in a web application in WP 6. The application, which was designed along with the development of the method, greatly shaped the underlying relationship model. The goal was to automate recurring and tedious tasks during the blueprint customization process spanning about 300 functional requirements. Especially in the light of visualizations based on minimal input the development of a web tool was deemed essential.

The development of the architectural components and blueprints was conducted by utilizing a Neo4J graph database which both offered a relevant tool to manageably capture and manipulate data during the development process and is machine readable for use in the web tool. Consequently, all relevant data needed to construct the blueprints as per chapter WP3 Architecture Blueprints (Lead FIR) and respective elaboration in the appendix is maintained in a database instance, hosted by research partner Sirris. Building on this, a Next.js application was designed to assist the user in the selection and blueprint customization process. As mentioned above, this framework relies on a definite state to derive results (generated webpages) from and incorporates trackable changes to that state, which greatly influenced the developed method's logic. The Next.js application is hosted on account of research partner FIR and is publicly available at bluesam-tool.fir.de without the need to register (any demand on persisting data was outsourced to the user via file down- and upload).

Manual on Applying the Method with the Webtool

After offering general information about the project context, the tool initiates the blueprinting process by requiring the user to select relevant use cases to be considered for smart product development (Figure 11). The use case selection is accompanied by elaborate descriptions of the use cases, available with a click on the respective info buttons. This stage also allows the upload of configurations, which were downloaded in a previous blueprinting process, to resume customization.

The selection of use cases should be oriented with a comprehensive goal in mind, like a set of use cases that are to be realized, or a set of use cases that are to be explored. This ensures that following architectural measures can be interpreted easier regarding actual development tasks or determining a potential delta to the individual as-is situation. Even though this stage allows for the selection of one up to all use cases, which in light of combined use cases does scale appropriately, for clarity purposes no more than three use cases are recommended to be selected simultaneously until the workings of the tool are familiar to the user.

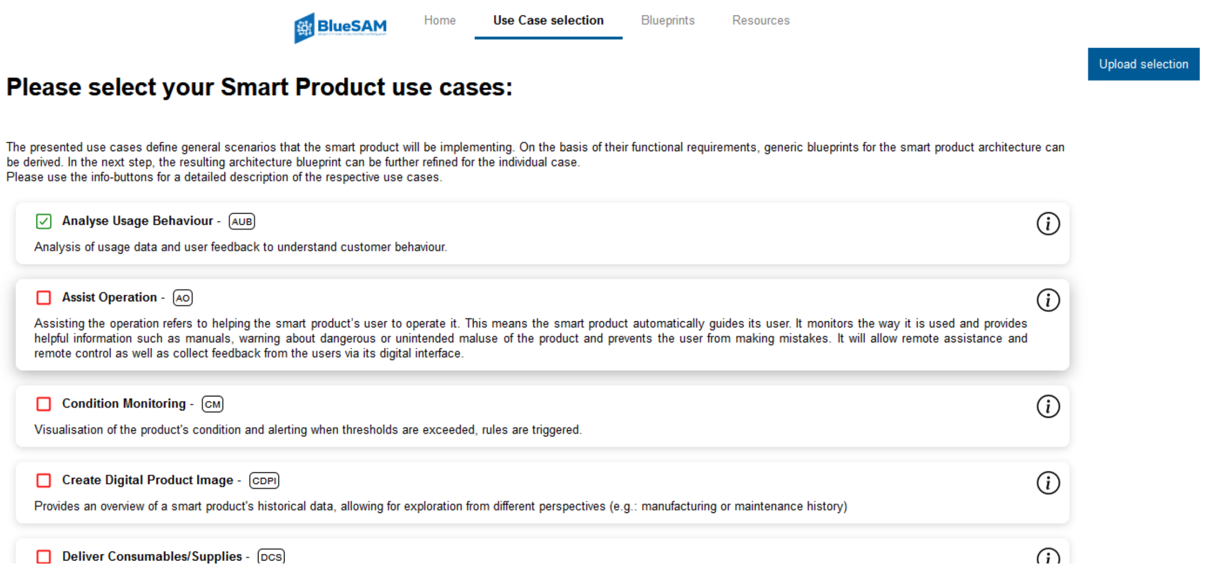

Figure 11 *Web tool - use case selection*

After the use case selection, the use can continue via the tabs at the top navigation bar or the "Discover the Blueprint" button at the bottom of the page. The tool presents (in case that no upload was conducted) the default (combined) blueprints of the selected use cases, starting with the functional view. Via the buttons on the right side, the user can switch between the different views, customize the blueprints via the functional requirements and export the state of the application to resume later.

The functional view (Figure 12) offers an overview of the functional elements for the blueprint grouped by their respective clusters. By hovering over the elements, a pop-up lists the constituting functional requirements and use cases (in case multiple were selected). This allows to retrace the construction of the view and to customize the requirements in the context of the individual applicability by toggling their relevance. In his way, it is possible to completely deactivate a functional element by eliminating all constituting functional requirements to it. Any changes to the functional view are automatically transferred to the other view.

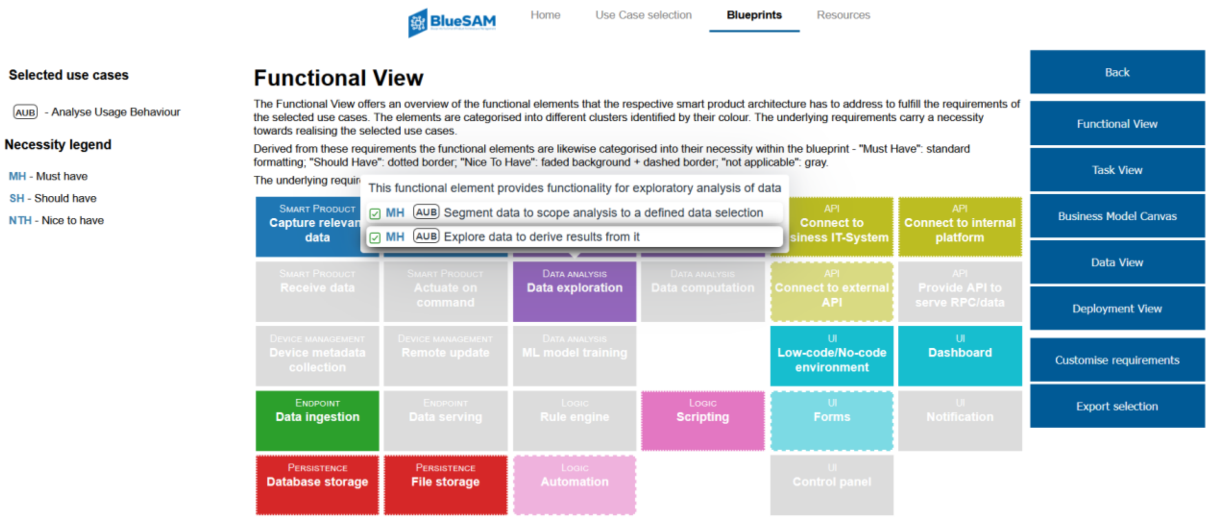

Figure 12 *Web tool - functional view*

The task view (Figure 13) displays the task elements, derived from the functional elements in terms of their domain and processual stage as described in the previous work package. Similarly to the functional view, the task can be inspected for underlying functional requirements by hovering over the elements. Furthermore, the visibility of the domain “edge” can be toggled via the checkbox above, to ignore it if it is not applicable for the individual case.

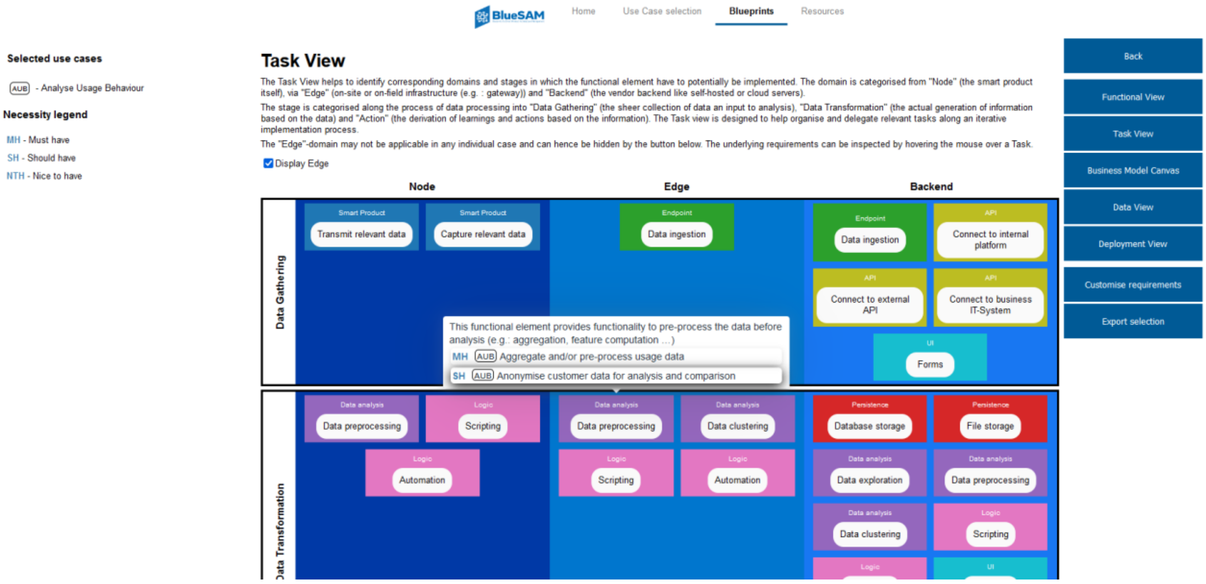

Figure 13 *Web tool - task view*

The data view (Figure 14) presents the possible relevant data objects that are to be considered during the development of the smart product. The clusters that the data element group into, are placed in a way that communicates a data flow from customer and public sources from the top to the data transformation and learnings below. With the checkbox above the illustration, the illustration can be toggled into a description of the clusters for further explanation on how to interpret the sections.

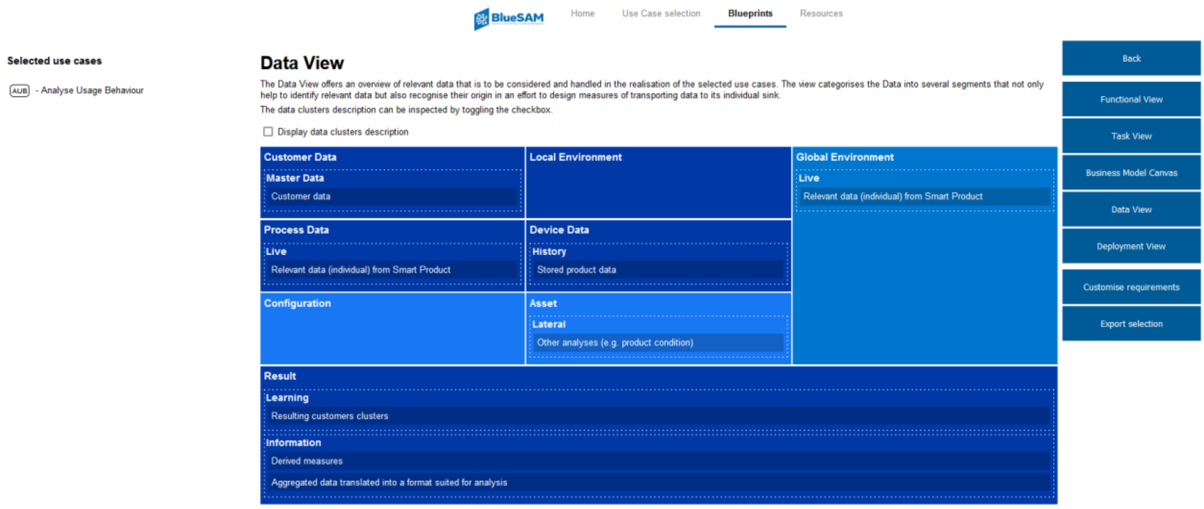

Figure 14 *Web tool - data view*

The deployment view (Figure 15) lists a collection of possible technologies by each functional element that they enable. The presentation format as a table allows to readily list and scope for related information like the deployment options and the skill levels. The illustration is designed in a way that the technologies are listed redundantly for each functional element but are hence easily accessed by looking for function specific tools. Below the table the contents from the chapter Additional content for blueprints – Architecture Management in the appendix are listed, to present a method for architecture management, where the selection of these technologies most likely will come into question. The intend is for the user to employ an active management in which the use of the presented technologies is being discussed and eventual further solutions pursued.

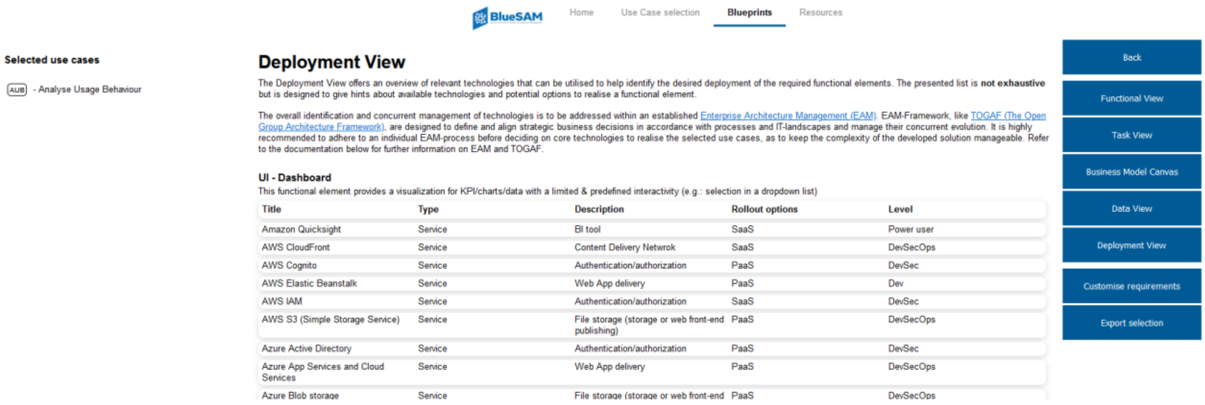

Figure 15 *Web tool - deployment view*

The business model canvas (Figure 16) combines all the business model elements from the use cases into a single canvas. The structure is identical to Osterwalder's Business Model Canvas to provide a familiar overview of relevant business aspects. The view is intended both to introduce business relevant questions into the process of architecture development and to allow for a critical re-evaluation of whether the induced scope of the selected use cases is consistent with the intended business strategy of the user.

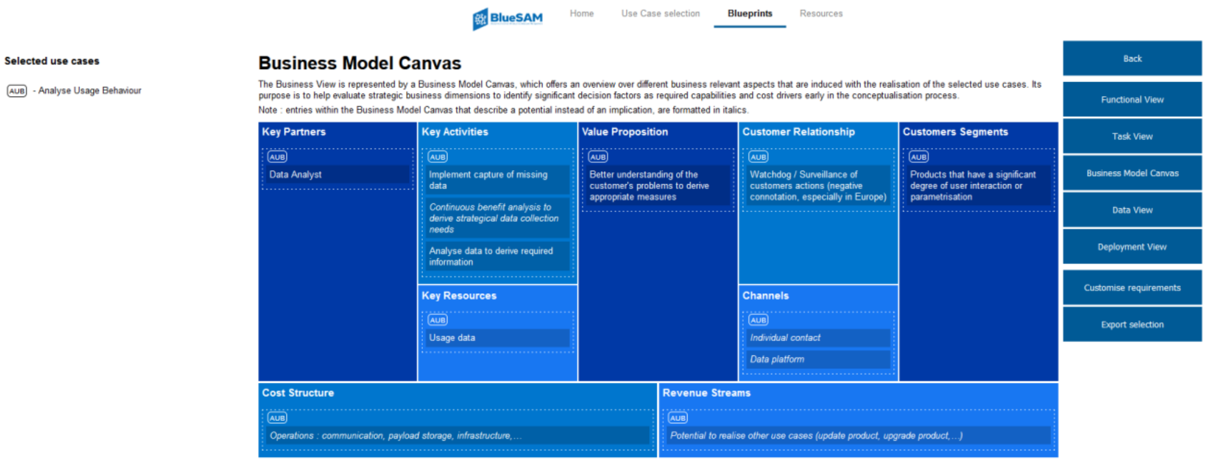

Figure 16 *Web tool - business model canvas*

The blueprint including all its views is derived from the use case related functional requirements, as explained above. To get an overview of the functional requirements and make direct changes to these for customization purposes, the button "customize requirements" will lead to a corresponding page (Figure 17). This customization page allows to assess the respective necessities of and directly filter for relevant functional requirements. Any changes are automatically transferred to all views of the blueprint.

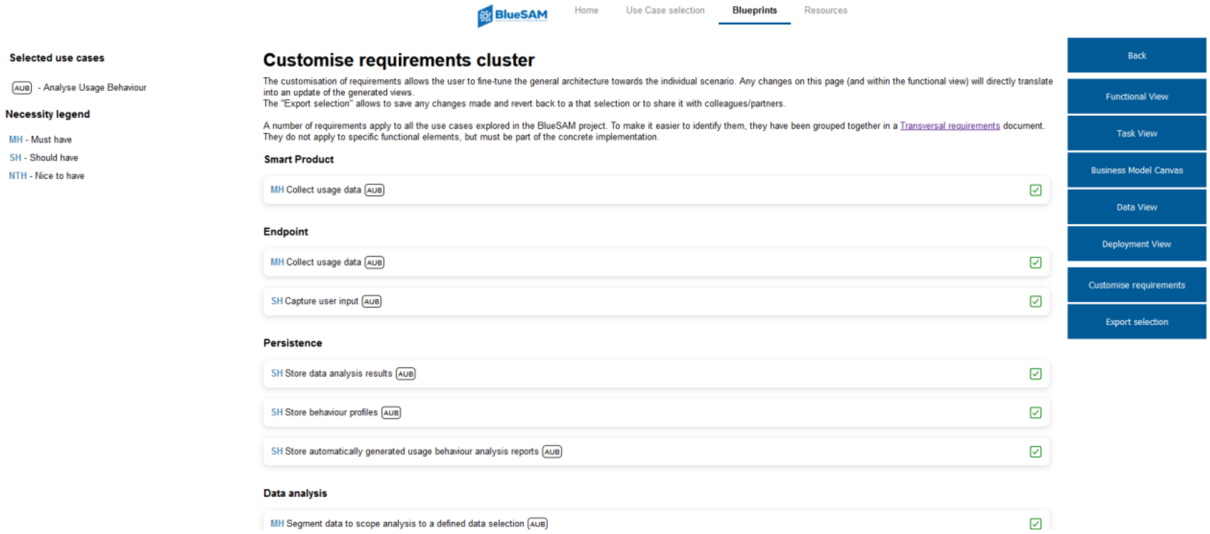

Figure 17 *Web tool - customization of functional requirements*

Finally, the resources tab in the navigation bar leads to further documents presenting the additional content. From the resources page, the contents for the transversal requirements, the elaboration on the deployment options and the co-development guideline can be downloaded in PDF format (see Figure 18).

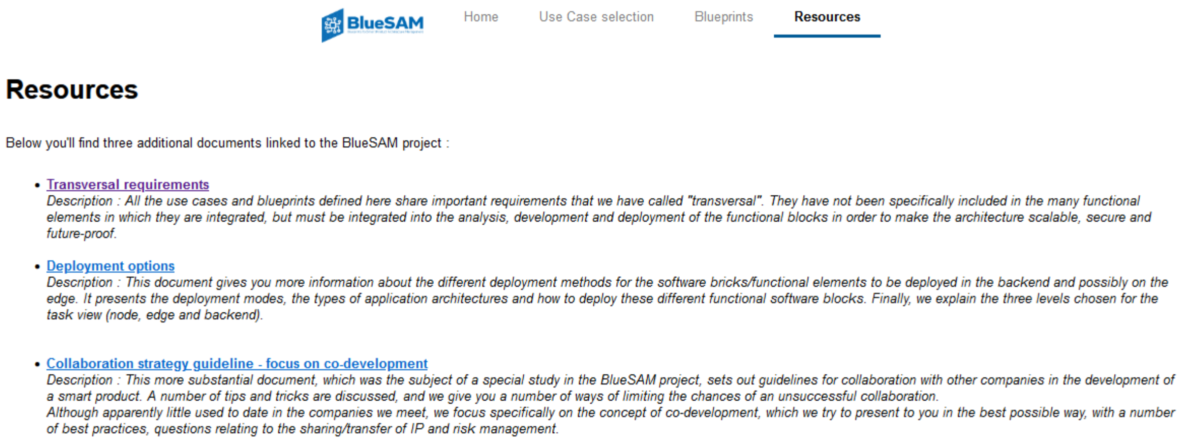

Figure 18 *Web tool - resources tab*

WP5 Co-Development Guideline (Lead Sirris)

The objective of WP 5 was to develop a guideline for SME on considering co-development as a viable option for developing smart products. The objective addresses the concern that SME do not have the necessary technical expertise to realize smart products and thus risking an advantageous market position due to a poor time to market. The goal of this work package is to demystify the process of co-development for SME, searching for solutions and highlight potential opportunities and pitfalls.

To investigate the topic of co-development broad literature research was conducted to identify relevant aspects. The first research was consolidated into a collection of 12 sources that address relevant issues to be collected into a first state of the art. The addressed issues comprise:

- Definition of the term “Co-Development”
- Varieties and degrees of co-development partnership
- Co-development strategy
- Approaching parties
- Legal agreements or rules in the partnership (especially intellectual property, IP)
- Benefits
- Success factors and best practices
- Risks and challenges

Having defined a baseline, of the matter of co-development, the consortiums addressed the user committee and additional companies in the research partners’ networks. In multiple interviews the interviewees were asked about their understanding of co-development and their experiences, motivation, and concerns. The interviews did incrementally shape the framework and identify relevant features of the to-be-developed co-development guideline. The final co-development guideline can be found in the chapter Additional Content for Blueprints – Co-development Guideline in the appendix and within the resource tab of the webtool. In addition to the written guideline, the overall learning from the interviews is that the main concern regarding co-development is the intellectual property (IP). In the committee’s experience, co-development is more often replaced by a service relationship, in which the legal aspects are easier to deal with. This is due to the practical nature of often dealing with an unequal relationship between a large and a small company, where the larger company can dictate the rules of engagement. By utilizing a co-development partnership, a smaller company might not have enough leverage in the negotiation to clarify, claim or protect IP. Therefore, although co-development does represent an attractive mode of operation in theory, in practice the burden on legal departments currently hinders the adoption of such a model. This problem statement is focused on in the developed guideline by sensitizing the reader to define a strategy to identify suitable partners and highlighting the requirement of clear communication of intended IP.

WP6 Demonstrator Validation (Lead FIR)

The objective of WP 6 was to validate and demonstrate the project’s main results developed in WP 3 and WP 4: the architectural blueprints and the blueprinting method. The goal was to demonstrate both the application of the method to yield a smart product and the resulting smart product itself, as an advertisement for the project contents. Additionally, WP 6 was set to develop a web application to host the contents of the BlueSAM project. As this development was closely coordinated with WP 4, the respective workload within WP 6 focused solely on the technical implementation of the web tool and is therefore not included in this report (basic information: Neo4J instance hosted by Sirris, Next.js application developed and hosted by FIR publicly at bluesam-tool.fir.de).

The development of a smart product for dissemination was originally intended to originate from a real-life product case from the user committee. In discussion with the production side of the committee it became clear that they either identified as already having a smart product in the field, or as having a solely non-digital product. Consequentially, the development of a smart product within the consortium would have required the realization of a technical base, which involves the scouting, purchase and installation of sensor and actor technologies, before any development of the empowering IT-Architecture would be applicable. Due to time-constraints of the project, the consortium decided to look for alternatives that apply for a swift application of the method and realization of the measures derived. As a result, research partner Sirris focused on the development of a meeting room environment assistant, for which a first technical concept

(IR-actor and microcontroller) was already available. Research partner FIR focused on finding an open-source espresso machine, that was already equipped with an extensive sensor and actor package, to finetune the machine's operation.

Demonstrator Sirris

The demonstrator developed by Sirris addresses the issue of indoor climate in an office environment. While air condition technologies are mostly usually present in the building's infrastructure, the solutions do not always provide the desired effect of an adequate climate or cause wasted operation during vacancies. To solve this problem, Sirris devised a smart product that implements the three following use cases:

- **Condition monitoring**: monitoring the room condition and visualizing historical data. Moreover, monitoring the device's condition.
- **Assist operation**: Web remote control of the air conditioning system through IR actuation on the smart environmental assistant
- **Optimize operation**: Proactively ask the meeting host if they want to turn on the AC system if the temperature is not ideal. Analyze the time it takes needed to reach the correct temperature to optimize the operating time.

Sirris started the development process with the definition of the use cases according to the methodology. The automatically attained blueprint for the use cases were then customized according to the actual functional requirements of the devised product. As a result, the blueprint's functional view resulted in the elements shown in Figure 19. (Although the final webtool was still under development at this stage, the digitally assisted method was already operational and therefore utilizable for the development. The figures for this report were generated with the final public version of the webtool).

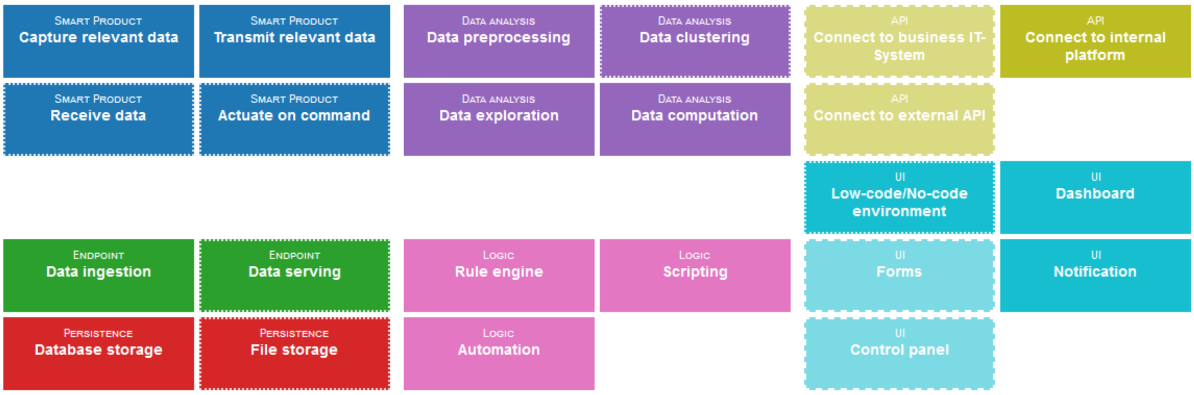

Figure 19 *Functional view for room assistant blueprint*

Moreover, the formation of functional elements resulted in a task view as shown in Figure 20. As the development strategy of Sirris devised a product that does not connect to an edge layer within the company, but acts as a cloud enabled device directly, the edge domain was omitted from the task view. Greyed out tasks represent those that due to the customization of functional requirements were deemed non-essential. By utilizing the technical view and a previously devised IT-strategy, Sirris defined technologies to enable these functionalities, contextualized by domain and processual (stage). As the preference was set onto a cloud service, which does not require its own infrastructure to run product or service logic, AWS was chosen as a cloud provider, as it incorporates all necessary functions. This matched with Sirris' skill level to set up and administer cloud environments and to introduce custom programming logic based on development frameworks rather than commercially available software solutions.

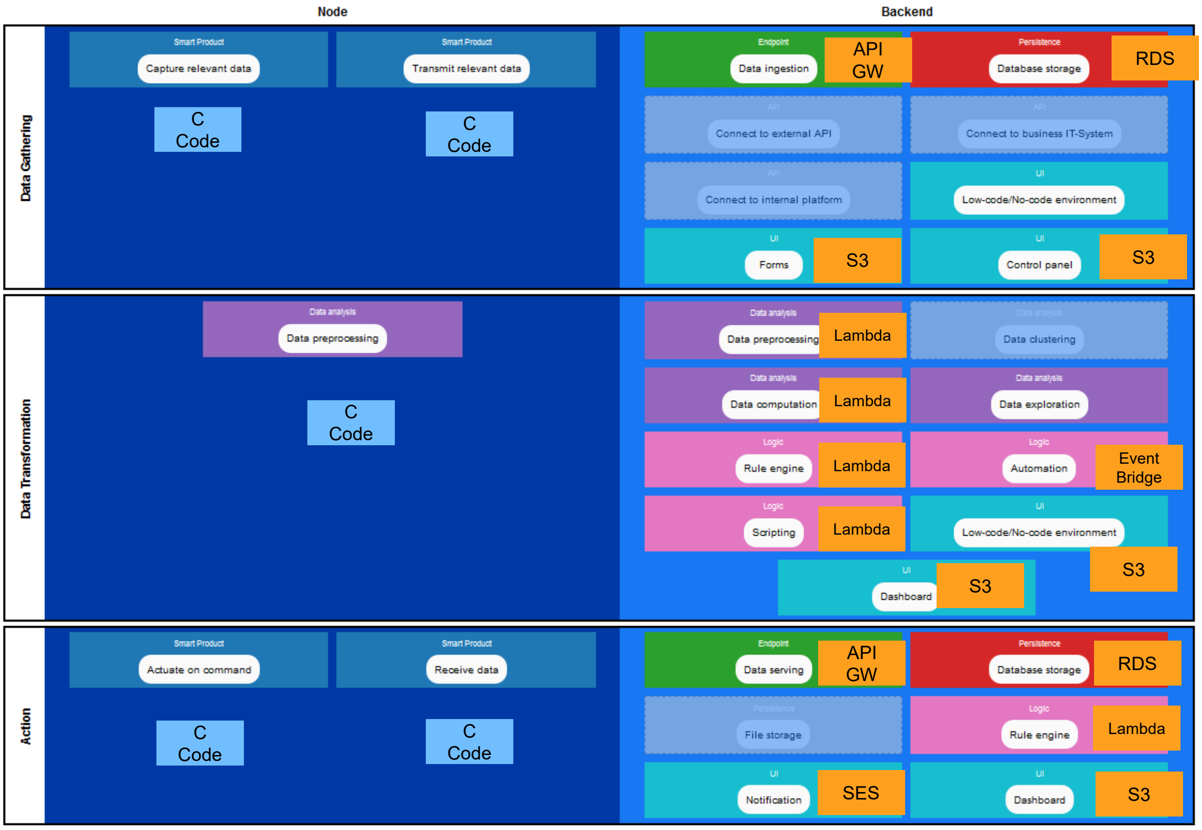

Figure 20 *Task view for room assistance blueprint including custom technical decisions*

Finally, the method allowed Sirris to define the products architecture (see Figure 21). The assistant device connects to the AWS platform via an API gateway to feed into logic and data analysis components realized with AWS lambda functions. The user interacts with the device directly via buttons or via a web interface hosted with AWS S3. The backbone is comprised of integrations into the companies' calendars, a public weather service, and a dedicated database for measurement data and configurations.

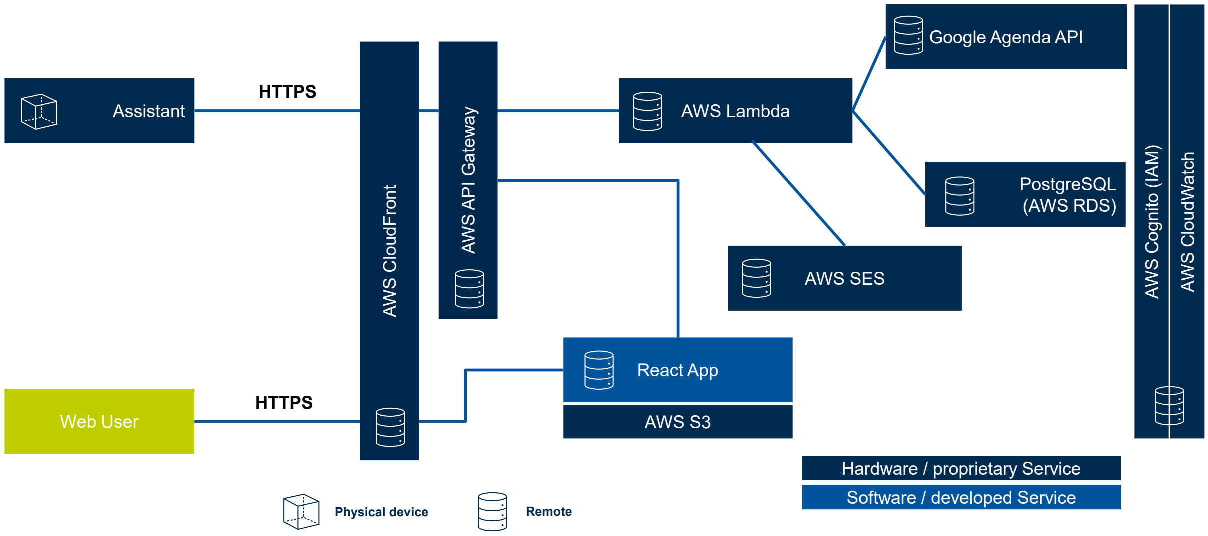

Figure 21 *Resulting architecture for room assistance demonstrator*

The final smart product, which is now defined not only by the physical device situated in each meeting room show in Figure 22, but also by the realized digital services implemented within the architecture. The room assistant does now allow personnel to not only control climate in meeting rooms, but also plan the climate ahead of the meeting either manually or automatically by synching the device to the organization's room booking. By capturing measurement data, the user is then able to finetune the control parameters and operating times.

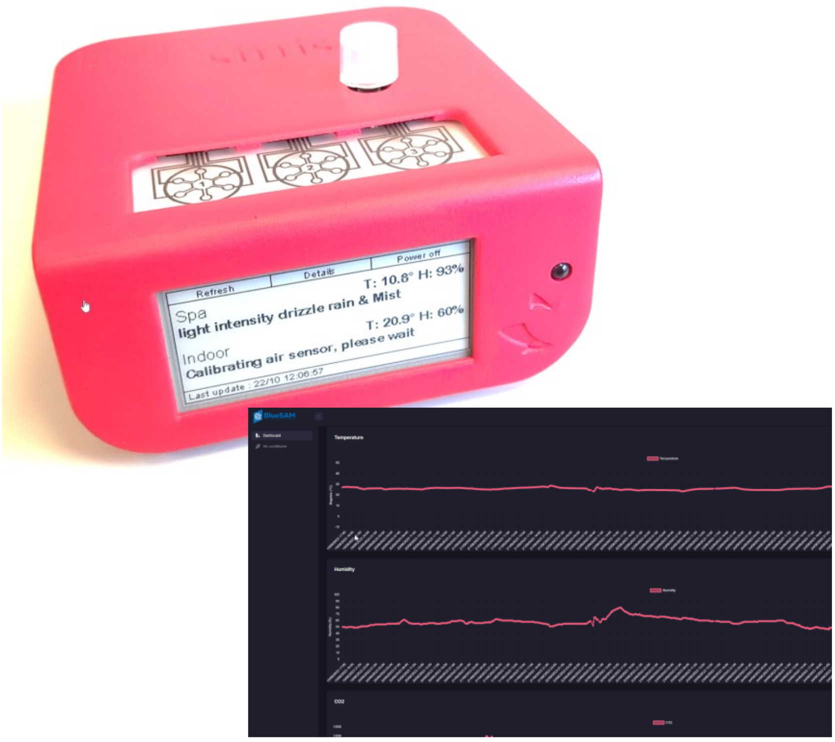

Figure 22 *Room assistant smart product demonstrator*

Demonstrator FIR

The idea for the demonstrator from FIR, originated from the prospect of having a mobile demonstrator for attending conventions in the domain of production systems and machines. In the context of a convention the attractiveness of the demonstrator was deemed more essential than an actual showcase of a production machine if a comprehensible analogy can be constructed. In the research for potential candidates, the research team stumbled upon the "Decent Espresso" espresso machine, which is a niche product for espresso enthusiasts (see Figure 23). The unique selling point of this machine is the fully digitalized machine which can not only capture process parameters of the brewing process, but also enables the user to actuate the process in a precise manner by constructing process profiles for the machine to run.

Figure 23 *Decent Espresso espresso machine as sold by the producer*

In discussion with the vendor, FIR learned that the machine was designed to be open source and hence open for individual customization. Also, the vendor stated that they do not intend to implement any smart product like features, such as data driven services, as their strategy is to provide the platform only. The combination of a fully digitalized, open-source machine without any implementation of smart product features was a good fit for the task at hand of building a demonstrator. In addition to technical aspects, the analogy between a digitalized portafilter machine and a production machine is straightforward to convey, as both employ a manual process by a worker, a machine-operation by the worker, and finally a parametrized, digitally controlled process.

Having decided on the technical basis for the demonstrator, FIR defined three use cases to be implemented and demonstrated with this machine:

- **Offer Data Analytics**: Allow users to inspect process data and accumulated usage data to gain experience in the operation of the machine and learn about other users' behavior.
- **Improve Product and Services**: (with FIR as role of the vendor) Identify critical process parameters and app functions by incorporating user feedback and studying captured data.
- **Assist operation**: Assist the user in the non-trivial process of machine operation to create better espressi

Again, FIR started the development process by defining the use cases, as per method. The automatically attained blueprint for the use cases were then individualized according to the actual functional requirements of the designed product. As a result, the blueprint's functional view resulted in the elements shown in Figure 24. Functional elements that were not selected due to the customization of the functional requirements are explicitly crossed out.

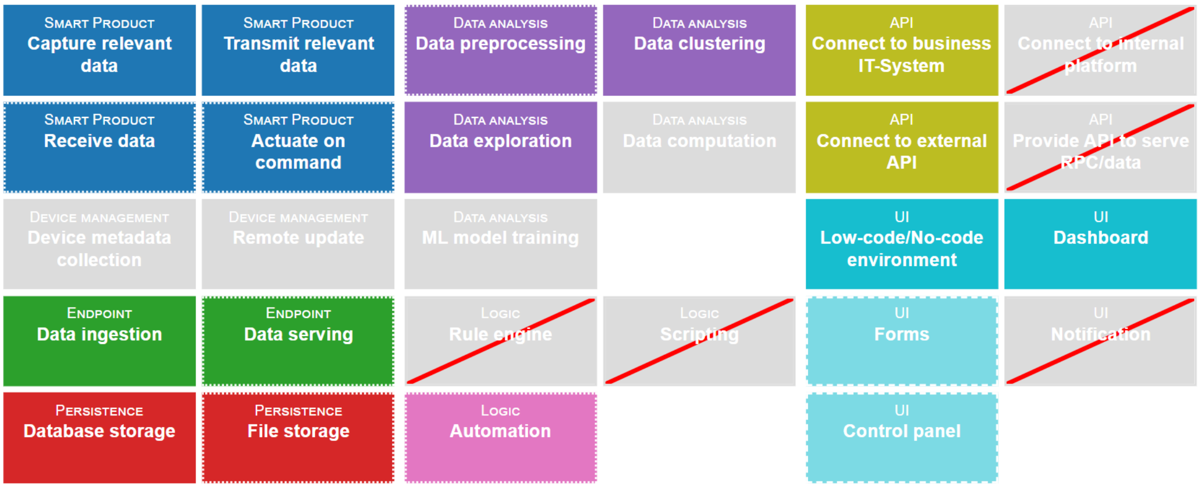

Figure 24 *Functional view for espresso machine blueprint*

The formation of functional elements resulted in a task view as shown in Figure 25.

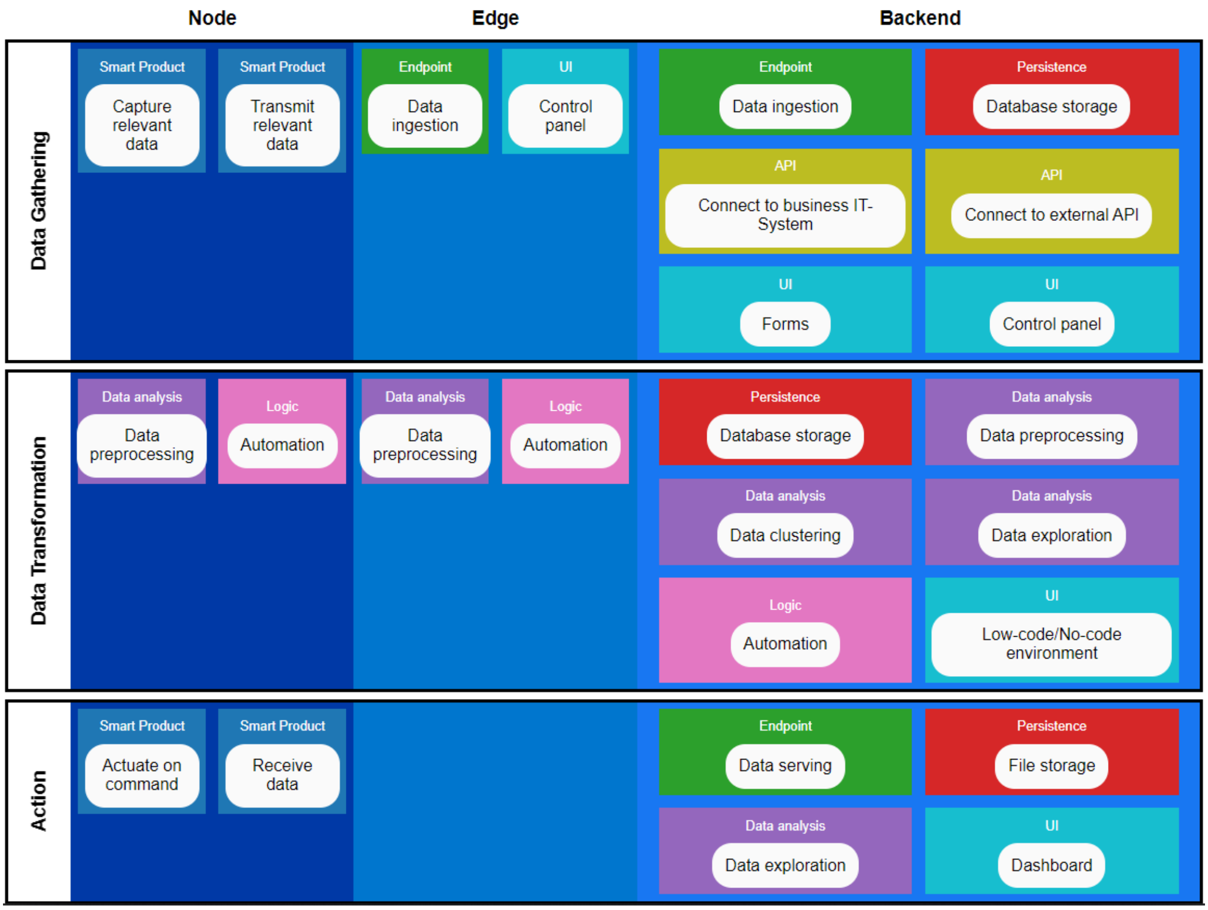

Figure 25 *Task view for espresso machine blueprint*

The technological consideration by FIR differed to Sirris approach, due to a different strategic approach. While Sirris' strategy envisioned the utilization of cloud services, the current research field of event-driven, and self-managed architectures by FIR required the architecture to be developed on-premises and under consideration of clearly specified domains. Even though the technological view of the blueprint could not contribute to this development (because no commercial products were planned to be utilized), the task view deemed to be a good tool, to structure the modules to be developed (see Figure 26). Hence any endpoint on the product side (“node”), namely the espresso machine, a Bluetooth enable scale, a USB-enabled NFC-reader, and a tablet for user interaction, was assigned with necessary functionalities defined by the tasks. An edge layer, consisting of a Raspberry Pi, was then assigned the remaining functionalities that could not be provided directly by the endpoints. The planned backend services were then structured according to the clusters of tasks. Some functionalities did not apply for the individual use case, but were not filtered out by the method, so a plausibility check step was added to remove the remaining, unneeded functionalities.

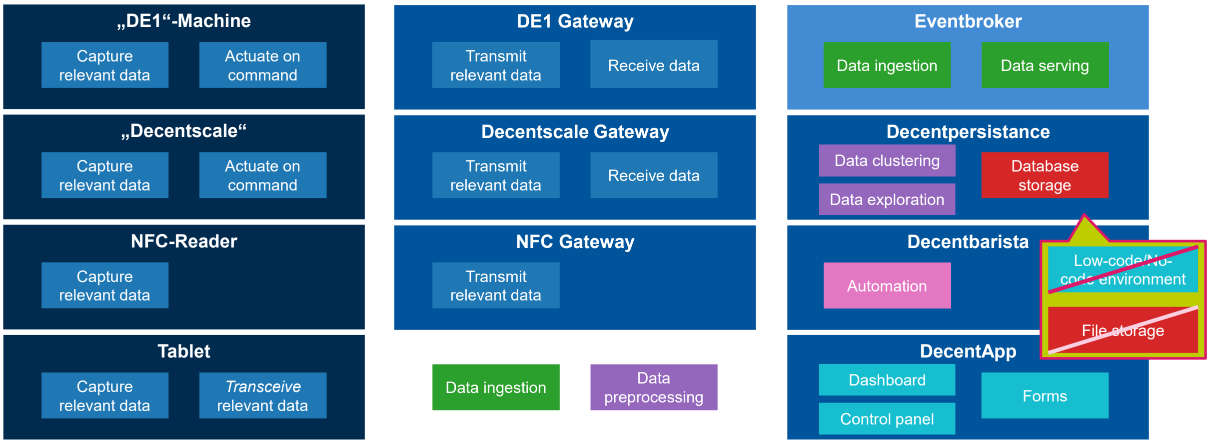

Figure 26 *Service structuring influenced by technological and strategical consideration for espresso machine*

More specifically, individual functionalities, such as a feedback function on the quality of the espresso, were by design not handled by the functional view, but could be evaluated using both the task and the data view. Anecdotally, this specific functionality was at first designed erroneously right after the espresso creation process. This did not consider when and where the information of espresso quality experienced by the user would be available. As the user will most likely leave the espresso machine and consume their espresso within the following next 5-10 minutes, the functionality was designed as an email to the user instead of a questionnaire directly on the machine’s tablet.

After defining and structuring the functionalities and data considerations of the demonstrator, the architecture was specified as shown in Figure 27.

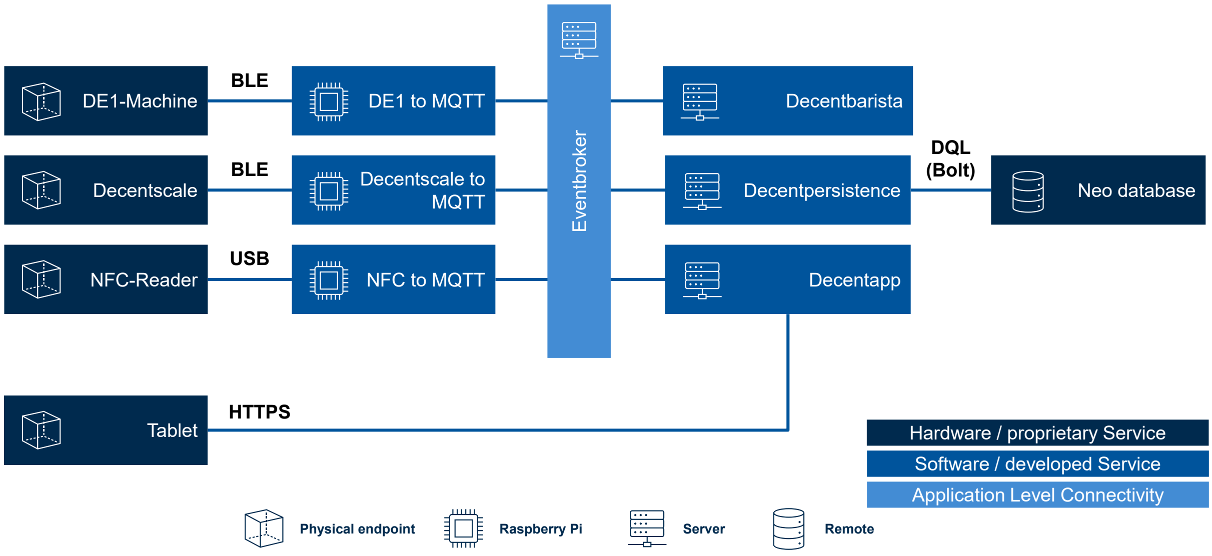

Figure 27 *Resulting architecture for espresso machine demonstrator*

The central communication platform is, as devised by an event-driven architecture, an event broker that handles communication between components in the form of events. A Raspberry Pi interfaces the Bluetooth (BLE) and USB interfaces of the espresso machine, the scale, and the NFC reader, and hosts device-specific services that act as a gateway and general interface to the devices on the event network. The remaining services are hosted on the backend, a server in the FIR infrastructure, and are structured by functionality. The “Decentapp” service hosts the user application that interfaces with the tablet on the machine. The “Decentpersistence” service interfaces a Neo4J database that provides persistence and data exploration capabilities regarding users, espresso profiles, shots, and measurement data. Finally, the “Decentbarista” service handles automation within the network, such as controlling the brewing process in response to the scale data and managing the handling of measurement data.

The resulting demonstrator is a mobile espresso machine that offers smart services to the user (see Figure 28). The assistance of operation is handled by both actively, during the espresso making process that instructs the manual operation of the user and stops the brewing process automatically in response to measurement data, and passively, by offering a wiki page in the user interface, teaching further information on the process of espresso making. Furthermore, the open architecture allows the remote operation of the machine because the user interface is hosted on the backend, which allows for direct assistance by experts. The data analytics use case is realized by presenting the user the measurement data of the brewing process, which allows them to assess possible errors, ratings by other users, to identify popular espresso profiles, and a temporal usage map, to identify popular times of operation and own consumption statistics. Finally, the use case of improving the product and services is addressed by using feedback functionality for the created espresso and the entire app. While the user feedback feeds into the espresso profile ratings for the users, it also allows to identify popular parameter traits in the profiles to develop new espresso profiles. Feedback to the overall app can be addressed individually, while the capturing of general measurement data allows developers to analyze the overall performance of the machine.

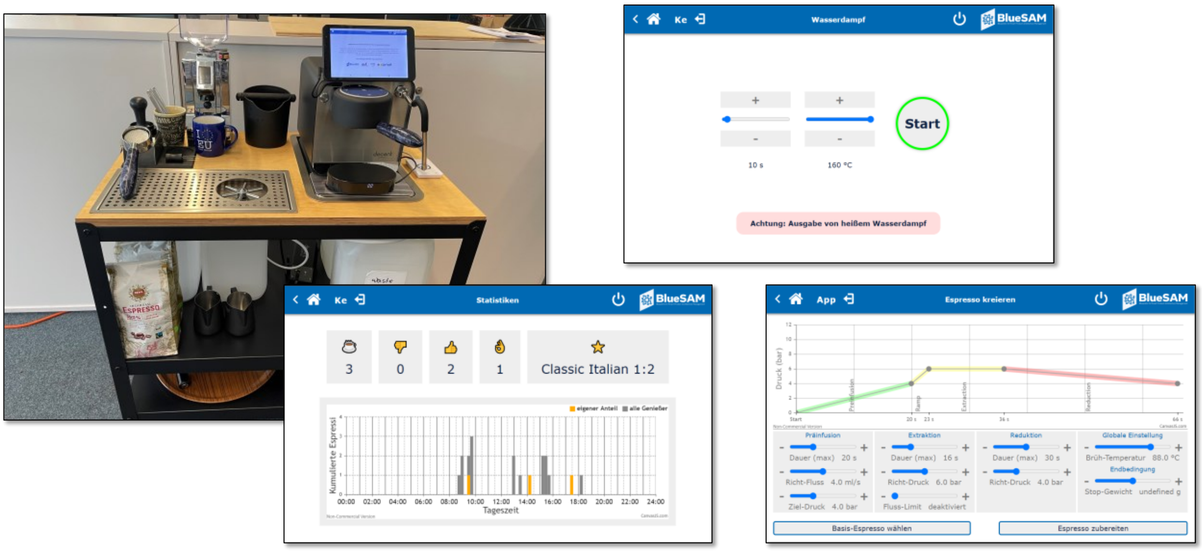

Figure 28 *Espresso machine smart product demonstrator*

Validation of Project Contents

The validation of the project contents was conducted by developing own demonstrators utilizing the method and blueprints. Although, the unavailability of eligible products in the user committee hindered the extensive validation of all contents, the smart products developed by the research partners offered valuable insights and learnings.

While WP 2 produced a reference architecture for BlueSAM, the design of the blueprints to be structured in different view was considered very valuable in the development process. As different companies employ different IT-Strategies, the blueprints are in no way eligible to prescribe the usage of cloud services or specific technologies. The functionality-, task- and data-oriented approach allowed for the subsummation of more abstract building blocks into an individualized architecture. Specifically, this was demonstrated by successfully utilizing the blueprint to derive an individualized architecture with oppositional strategies: Sirris focused on a cloud- and platform-based approach and FIR on a self-managed, open architecture approach. This result reinforced the view-based approach in the blueprints, with the absence of a technology-specific building block.

The customization of the blueprints via the functional requirements allowed both Sirris and FIR to individualize the blueprint results to focus on the area of concern. However, this approach does not produce final results and needs to be understood accordingly. Both Sirris and FIR had to fine-tune the selection of functional, task and data elements, due to the specific requirements of each smart product that could not be generalized by the tool. This emphasizes the general and more exploratory nature of the method developed, which does not create ready-to-develop solutions but must be understood as a tool to structure the beginnings of a development process.

Since the research partners did not come from a business-driven smart product case, the evaluation of the blueprints' business view, i.e., the business model canvas, could not be extensively conducted within the development of a smart product. However, any irregularities perceived during the development process were addressed directly by adjusting the contents in the Neo4J database.

Regarding the overall method, the finished web tool was used by some members of the use committees in a productive interest. The research partners accompanied the design process and helped with the overall operation of the tool. Although specific information cannot be disclosed in this report, the use of the tool was generally considered beneficial to the development. In particular, not all views were of interest to each single user. Yet the preferred view of the tool varied greatly from user to user. The task view was deemed valuable for its allocation of functionalities across the processual stages. The data view provided a beneficial structure to discuss specific data points across the smart products environment. The deployment view offered a good overview of partly unknown technologies to be considered for implementation. Overall, each user could identify a beneficial quality of the method, to highlight an individual perception of the task at hand.

Dissemination and Creation of a Workshop

The development of the demonstrators directly by the research partners improved the dissemination capabilities immensely. Especially, the espresso machine as a demonstrator for smart products has, as expected, increased interest in the project. While one of the initial goals of WP 6 was to develop a workshop format, the demonstrator itself proved to be valuable and effective in opening the dialogue on smart products. Thus, the devised workshop was designed towards a slide deck about a general motivation as to why IT-architecture needs to be addressed, with the demonstration of the espresso machine shifting the focus to smart products; finally, the method behind the development of such a smart product is highlighted and explained in the scope of the web tool, for the participants to try out live in the workshop/meeting under the supervision of experts.

This mode of workshop was first conducted during the final meeting with the BlueSAM user committee, where the content motivated the participants to continue discussion after the event. With the creation of additional material and in the context of other thematic focuses (like upgrade circular economy, enterprise IT-architectures or connected shopfloors) the (respectively adapted) workshop format was carried out at events such as:

- Project marketplace of the Center for Integrated Business Applications at FIR
- Digital Transformation Expert certification course at FIR
- Mittelstand-Digital Roundtable (hosted by user committee member Ph-Mechanik)
- Zuse community assembly in Berlin (Figure 29)
- Roundtable Information Technologies (online) by FIR
- EDIH Rheinland TAKE-OFF at digitalHUB Aachen (Figure 30)
- Maschinenbau-Netzwerk visit at FIR

Figure 29 *Demonstration of BlueSAM at assembly of the Zuse community*

Figure 30 *Demonstration of BlueSAM at EDIH Rheinland, published by Lokalzeit Aachen on public broadcast*[1]

[1] Short report available at https://www.ardmediathek.de/video/lokalzeit-aus-aachen/lokalzeit-aus-aachen-oder-24-10-2023/wdr-aachen/Y3JpZDovL3dkci5kZS9CZWl0cmFnLXNvcGhvcmEtNzVlMTU5NGEtNjEyNC00YmM1LWE5NWItMjEzMGY2MjAxNDI1

Usage of grant

The expenditure of scientific and technical personnel shown in Table 7 was required to process the individual WPs.

Table 7 *Overview of the PM actually incurred by scientific staff within the research project*

Period	PM
2021	15,11
2022	12,69
2023	1,10
Total	28,90

Necessity and Appropriateness of the Work Performed

The work carried out in the BlueSAM research project is fully in line with the approved application and was therefore necessary and appropriate for the implementation of the project. It should be noted that the work in work package y had to be adapted as a result of the knowledge gained in work package x. The project schedule is based on the work diagram of the research proposal. The duration of the work packages was increased by the effort-neutral extension. The respective procedures in the individual work packages were discussed and agreed with the members of the project support committee.

Innovative contribution and benefits for SMEs

Innovation Target and Application of Results

The overall innovation target for BlueSAM was to speed up and support the development of smart products by SMEs. This was achieved by developing:

- a smart product reference architecture based on existing architectures and standards
- use-case-based architecture blueprints developed with SMEs,
- a method that enables to define and fine-tune a blueprint for digital architecture for smart products, tailored to the needs of an SME,
- a guideline for the co-development of smart products with technology partners.

In the beginning, an existing reference architecture was selected as reference architecture framework. Building on an existing framework will allow easier transition of the results. Its central building blocks and the main elements and modules were used as a first basis. Based on the reference architecture framework and the functional requirements derived from the case studies, a BlueSAM-specific architecture, tailored to SMEs, was composed to ensure both practical usability (use-cases) and conformity with existing standards (reference architecture frameworks). For each use case, architecture blueprints were then defined and detailed in different views, namely: functional, task, data, deployment (including technology & skill), and business.

The developed method and resulting blueprints address more concretely the following sub-objectives:

- To support a pragmatic approach that bridges the gap between the complexity of available reference architectures and an optimized blueprint for a specific SME to initiate envisioned use-case(s) or new business models.
- To start with an existing architecture and best practices allowing to use and build further on available building blocks, enabling rapid implementation and lowering an SME's own development effort.
- To guide SMEs in the assessment and realization of the envisioned smart product.
- To facilitate co-development with technology partners and suppliers, supporting efficient implementation keeping in mind the required time and related costs.

As such, the project results bridge the current knowledge gap at SMEs to initiate the development and successful introduction of smart products and services.

In essence, the innovation goal was achieved by addressing the lack of support for SME on the base of a proven and repeatable architecture definition through a SME use case-oriented methodology, while taking into account business and co-development aspects. This method will enable SMEs to avoid this reality of "smart product paralysis" (i.e. being blocked by the abundance of opportunities and decisions to be taken, as identified by Sirris in Belgium SMEs), encountered by SMEs wishing to launch themselves.

Economic Impact

The economic impact to the project's conclusion corresponds to the state of knowledge at the proposal phase: The digitalization of the economy forces SME machine, equipment, and tool builders to optimize their internal and external processes and to provide additional digital services to create and capture more value for their customers. These project results will guide SMEs through the first steps to turn their smart product ideas into professionally designed products, with high impact that can be produced on a large scale. Building on existing reference architectures, the process will enable the SMEs to develop future services and products based on their solution. The process and methodologies will help the SMEs to:

- Make informed decisions about the architecture of their smart product and co-development with technology partners:
    - Reduce the risk of project failure or excess in time and costs,
    - Reduce unnecessary costs and time for errors in architecture design [13,21]
    - Time-to-market reduction due to repeatability and structure of reference architecture and blueprints,
    - Reduce cost and time during the whole project due to co-development approach [22]
- Develop new smart solutions with a high quality, on time and within budget:
    - reduce lead time [23],
    - Improve product development in general time and cost [24],
    - improve product quality [25]
- Throughout the product lifecycle:
    - Realize new business models, leading to new markets [26]
    - Increase functionality,
    - Improve resource efficiency, reduce down-time, improve service quality [27]

The cost of the introduction in SMEs will be limited as the method is based on virtual concepts such as architectures and models. As the method and the guideline are provided on a web tool, the adaption effort will be small compared to its benefits. Access to the blueprint library will be free of charge for the SME and the method will enable them to make the best decisions while saving time and reducing risk. In general, the overall potential economic impact is regarded as significant [26].

Publications and Transfer Measures

User committee

The involvement of the user committee served to ensure the practical relevance and dissemination of the results. The committee meetings generated the desired interest from the business community. Although the participation of the user committee did not fully meet expectations towards the end of the project, other companies from the network of research partners could be involved to make up for deficits. By integrating workshops on the project into a working group of the FIR and by recruiting participants for the survey, it was possible to involve a sufficient number companies in the preparation of the results.

The companies officially involved are listed in Table 8. Additional participants are listed in the fundee's report on project-related (virtual) expenditures.

Table 8 *User Committee as per beginning of the project*

Company (incl. legal form and address)	Contact Person (delegate who will attend the meetings)	SME (X)
Yamabiko Europe SA Avenue Lavoisier 35 1300 Wavre	Pierre Desneux	
Inductotherm Coating Equipment SA Rue de Hermée, 197 4040 Herstal	Laurent Watteyne	
BBK SA (Biebuyck) Chaussée Paul Houtart, 160 7110 Houdeng-Goegnies	Yannick Ruffel	X
Stûv SA Rue Jules Borbouse, 4 5170 Bois-de-Villers	Thomas Duquesne	
A.M.B. Avenue Nicolas Copernic 1 7000 Mons	Frédéric De Meulemeester	X
Eloy Water Rue des Spinettes, 13 4140 Sprimont	Olivier Eloy	
SULZER Bois La Dame, 4 4890 Thimister-Clermont	Benoît Martin	
PAM Concept SPRL Rue Bastin 228 4020 Wandre	Philippe Amato	X
Travelec Première Avenue 215 4040 Herstal	Yannick Reichling	X
Microsys lab – University of Liege Rue du Bois St Jean, 15-17 4102 Seraing	Philippe Laurent	

Company (incl. legal form and address)	Contact Person (delegate who will attend the meetings)	SME (X)
MICROMEGA – DYNAMICS SA Rue du Trou du Sart 10 5380 Noville-Les-Bois	Jean-Philippe Veschueren	X
Granutools SPRL rue Jean Lambert Defrêne 107 4340 Awans	Filip Francqui	X
PEPITe SA Rue Forgeur 4 4000 Liège	Philippe Mack	X
Go2BE SCRL (Sweet Move) Rue L. Brecht 5 4840 Welkenraedt	Gaëtan Hanen	X
Electronics Engineering – Consult & Design (EECD) SRL Priesville, 5A 4845 Sart-lez-Spa	Xavier Breuer	X
EICe Aachen GmbH – Center Smart Services (CSS) Center Smart Services **Campus Boulevard 55** **52074 Aachen**	Benedikt Moser	X
grandcentrix GmbH Holzmarkt 1 50676 Köln	Christian Pereira	
HAHN Digital GmbH **Frankfurt Airport Center 1** **Hugo-Eckener-Ring** **Gebäude 234, 8. Stock, HBK19** **60549 Frankfurt**	Martin Bleider	
i2solutions GmbH **Pfarrer-Gau-Str. 13** **52223 Stolberg**	David Bergstein	X
Janz Tec AG **Im Dörener Feld 8** **33100 Paderborn**	Markus von Detten	X
mimatic GmbH Westendstraße 3 **87488 Betzigau**	Karl-Heinz Schoppe	X
NEAC Compressor Service GmbH & Co KG Werkstraße 52531 Uebach-Palenberg	Stefan Damberg	
Oculavis GmbH Vaalser Str. 259 **52074 Aachen**	Martin Plutz	X
Ph-Mechanik GmbH & Co. KG **Güner Winkel 3-5a** **52070 Aachen**	Oliver Schürings	**X**
Zentis GmbH & CO. KG Jülicher Straße 177 52070 Aachen	Phillip König	

Company (incl. legal form and address)	Contact Person (delegate who will attend the meetings)	SME (X)
3WIN Maschinenbau GmbH **An der Schurzelter Brücke 11** **52074 Aachen**	Daniel Kaußen	X

Dissemination Plan during Project

Planned Activity	Objective	Realised Measures
Kick-off meeting with partners and user committee	Present the upcoming tasks and the work packages, discussion about the potential cases	**Kick-off meetings 26.05.21 & 27.05.21**
Project presentation (online publications on partners' websites and in newsletters)	Introduce project at targeted SMEs: needs, objectives and expected results	**FIR project page and continuous publication in FIR network newsfeed**
User committee meetings	-Ensure the technical and economic relevance of the project -Presentations, reports and discussions of the results and status of the project -Selection of cases -Presentation of achieved test cases	**UC meetings - 26.05.21 - 27.05.21 - 29.11.21 - 20.01.22 - 14.09.22 - 29.06.23**
User committee meeting	Final project results and conclusions	**Project conclusion at FIR 29.06.23**
Workshop with demonstrators	Demonstrate the interest of the developed methodology through concrete examples	**During Project conclusion at FIR 29.06.23**
Visit to companies/online meetings	Check relevance of topics and ask for feedback	**Open meet and greet during UC meeting 29.11.21; Guided tour of Demonstration Factory Aachen on 29.06.23**
Technology watch about the project subject and publications in Sirris media (Techniline, newsletters…)	Inform SMEs about the evolution of the technology	**Continuous in Sirris network newsfeed**
Project Website	Easy Information access for all stakeholders	**FIR & Sirris project page**
Press releases via IDW – Informationsdienst der Wissenschaft	Publications and Public Relations	-
Publication of scientific and practical articles in the FIR magazines "UdZ Forschung" and "UdZ Praxis"	Information for SMEs and potential users	**Publication of BlueSAM in UdZ issues 2.21, 1.22, 3.23**
Publication of project results on the FIR conference "Convention on digital opportunities – CDO" in 2020 and 2021	Presentation of project results to industry experts	**Inclusion in contents of CDO in 2021 and 2022**
Publications of project results in expert media such as IT&Production and Business & IT	Publications addressed to industry experts and SMEs	***The publication was changed into a third conference paper***

Planned Activity	Objective	Realised Measures
Presentation at VDMA / Bitkom Meetings	Publication of project results and exchange in expert networks	**Inclusion of project contents and learnings in:** - **Roundtable Information Managers (2021)** - **AiF research advisory board (2023)** - **Roundtable Information Technologies (2023)**
Presentation at two international conferences such as: CIRP CMS, IFIP, APMS	Publication of project results in the scientific community	**Accepted papers at:** - **ProVE 2022 [28]** - **CPSL 2022 [20]** - **CPSL 2023 [29]**
Two Best-Practice Exchange Meetings between Belgian and German UC-Members	Best Practice Exchange and transnational community building	**Open meet and greet during UC meeting 29.11.21**
Writing blog articles	Present interim results and raise awareness about the project	***This measure was shifted towards publication in own network newsletters and LinkedIn posts***
Visit to UC smart product companies	Raise awareness and knowledge through guided tours	**Guided tour of Demonstration Factory Aachen on 29.06.23 Interviews with Zentis, grandcentrix and Janztec**
Integration into university teaching	Seminar, Master and Bachelor theses on “Smart Products Architecture” integration into the lecture Production Management II	**Integration of project learnings and contents into certificate course Digital Transformation Expert**
Use in potential workshops and design workshops	Test/validate our methodology/ raise awareness about the project	**Creation of a presentation/workshop deck in unison with smart product demonstrator**
Presentation of the results at conferences and trade fairs	Further dissemination of the approach and tools developed	**Presentation at** **CBA 2023 (FIR, Aachen)** **CDO 2022 (FIR, Aachen)**
Webinar series “how-to Smart Products”	**Introduction and training on BLUESAM results**	***Need for webinar series has been substituted for availability of own demonstrators to present contents on***

Dissemination Plan after Project

Activity	Objective	Realised Measures
Publications in journals and periodicals such as IT&Production, Business &IT	Increase awareness of the developed tools and methodology	**Publication in** - **CPSL 2023** - **UdZ 3.23**
Service to SMEs	Support SMEs during the development of their smart products	**Assistance of 2 SMEs in the application of the webtool in private meetings**
Use of project results in educational activities	Dissemination of results through academic courses	**Continuous inclusion of project contents and learnings into certificate course Digital Transformation Expert**
Develop new training offer	Transfer of the project results into the training concept for the qualification of employees from SMEs	**Uptake of smart product conceptualization into FIR's industry portfolio spanning organizational and technological focuses**
Integration of the results into university teaching	Dissemination of results and integration into training; integration into scientific debate	**Results were used in Lecture Production Management A + B and Produktionsmanagement I + II**
"Masterclass" (Experts workshop)	Presentations of the developed tools and the methodology	
Workshop with the demonstrators	Demonstrate the interest of the developed methodology through concrete examples	
Presentation of the results at conferences and trade fairs	Further dissemination of the approach and tools developed	

Activity	Objective	Realised Measures
Publication of the project results on the Internet and web-platform in specialist magazines	Dissemination of the results within the relevant target group	**Presentation of demonstrator, method and workshop at:** - **Convention on Production Systems and Logistics CPSL 2023 (South Africa)** - **Project marketplace of the Center for Integrated Business Applications at FIR (2023)** - **Digital Transformation Expert certification course at FIR (2023)** - **Mittelstand-Digital Roundtable (hosted by user committee member Ph-Mechanik, 2023)** - **Zuse community assembly in Berlin (2023)** - **Roundtable Information Technologies (online, 2023)** - **EDIH Rheinland TAKE-OFF at digitalHUB Aachen (2023)** - **Maschinenbau-Netzwerk visit at FIR (2023)** - **IoT meeting (TechnoCampus) (2023)** - **Workshop : “Intégration de l’IA, de l’IoT et de la réalité augmentée au sein des activités du Sirris Product Development Hub” (2023)** - **Webinar : “Expanding your products with IoT” (2023)** - **Workshop : “Construire une solution IoT articulée autour de votre produit” (2023)** - **Sensibilisation à l'IoT, IA, Additive manufacturing, jumeaux numériques (2024)** -
Creation of a dissertation based on the project	**Dissertation based on the project results as great addition of the scientific contribution of the project**	**Project Results are used in Dissertation “Information Systems Architectures for Smart Products in Mechanical Engineering” from Team Member Max-Ferdinand Stroh. Planned Publication: End of 2024**

Assessment of the Feasibility of the proposed and Updated Dissemination

The proposed initial dissemination differs from the current plans, as the learnings during the project and the achieved project results provide new and different opportunities from those envisioned in 2021. With the development of an “own” smart product that was focused on the attractivity for dissemination activities, the focus of dissemination is now exclusively on the incorporation of the physical demonstrator.

This has already been perceived as a more effective dissemination measure than the envision online workshop course, as the participants can physically interact with the product, understand the projects contents on a practical level, and naturally learn about the difficulties in development motivating the use of the method, i.e. webtool. This allows for a more natural conversation about the project contents, without conveying tightly packed workshop contents.

Therefore, the updated dissemination activities, based on the physical demonstrator, are evaluated as a very effective method to transfer the project results. Finally, the results of the project will conclude in a PhD thesis at FIR at RWTH Aachen University.

Project Consortium

FIR e.V. at RWTH Aachen University

FIR, the Research Institute for Industrial Management at RWTH Aachen University, is both a research and technology organization and a non-profit association. It is engaged in the development of models for the organizational basis of the digitally integrated enterprise of the future by strongly involving industrial partners in the research process. The FIR develops models, methods and concepts for digitalization and smart products, which are implemented in cooperation with project partners, in both industrial and service enterprises in Germany and throughout Europe.

Sirris

Sirris is the collective center of the Belgian technology industry. Companies in the target group of this project are members of Sirris. The main role of Sirris is to help the industry, and notably SMEs, which are an important part of the industrial network, to develop their activities thanks to innovation. Sirris has a long history of research and technology transfer in the field of product development, with a recognized expertise in ICT, materials, product development, advanced manufacturing, mechatronics and sustainability. The technological domains and associated expertise covered by Sirris are in the fields of product development, advanced manufacturing, mechatronics and information technologies.

Funding Information

The IGF project 303 EN of the research association FIR e. V. at RWTH Aachen University (Research Institute for Rationalization), Campus-Boulevard 55, 52074 Aachen, was funded by the AiF within the framework of the program for the promotion of joint industrial research (IGF) program by the Federal Ministry for Economic Affairs and Energy on the basis of a decision by the German Bundestag.

We are grateful for the funding and support.

Appendix

UML-diagrams for smart product use cases

UML-diagram for use case Analyze Usage Behavior

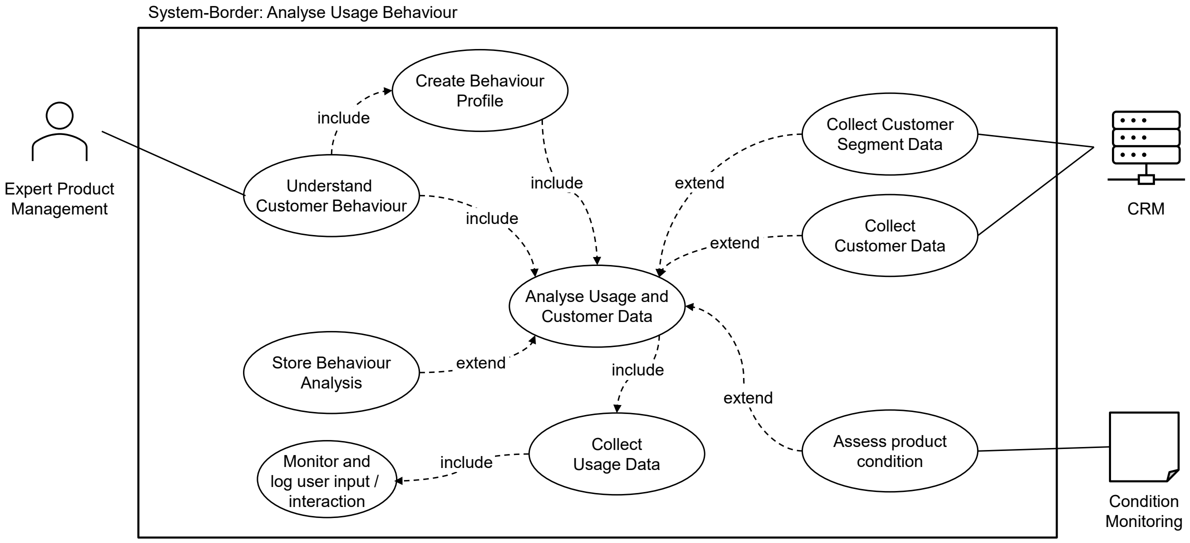

Figure 31 *UML-diagram for use case Analyze Usage Behavior*

UML-diagram for use case Assist Operation

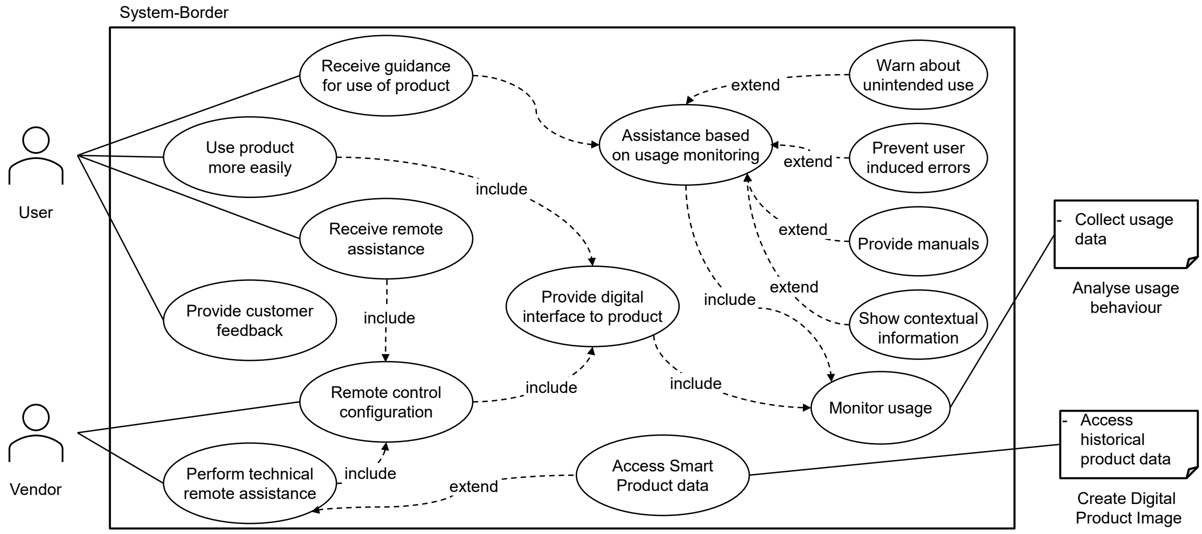

Figure 32 *UML-diagram for use case Assist Operation*

UML-diagram for use case Condition Monitoring

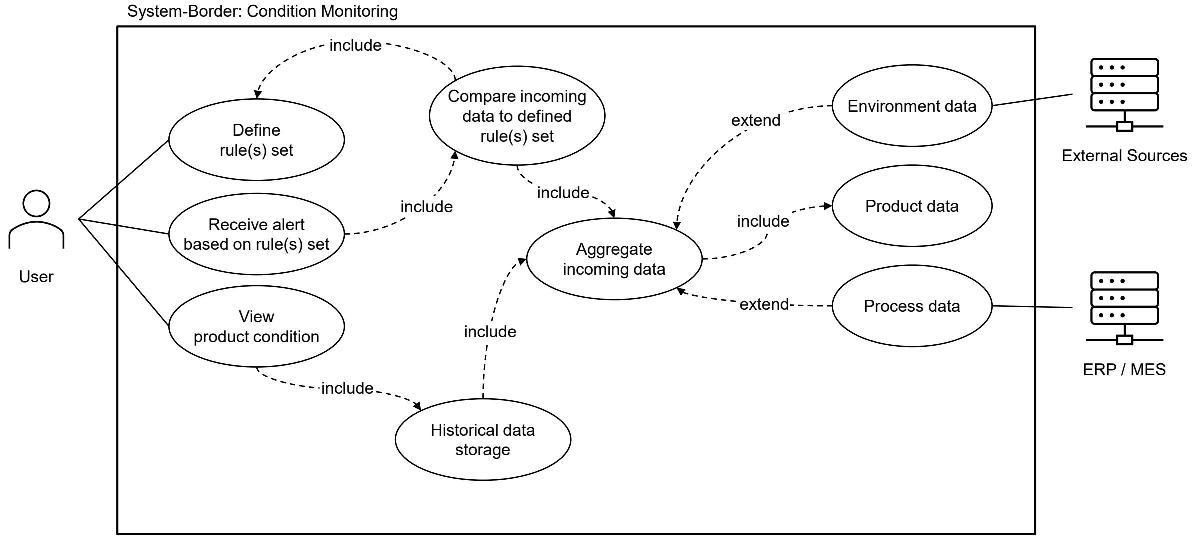

Figure 33 *UML-diagram for use case Condition Monitoring*

UML-diagram for use case Create Digital Product Image

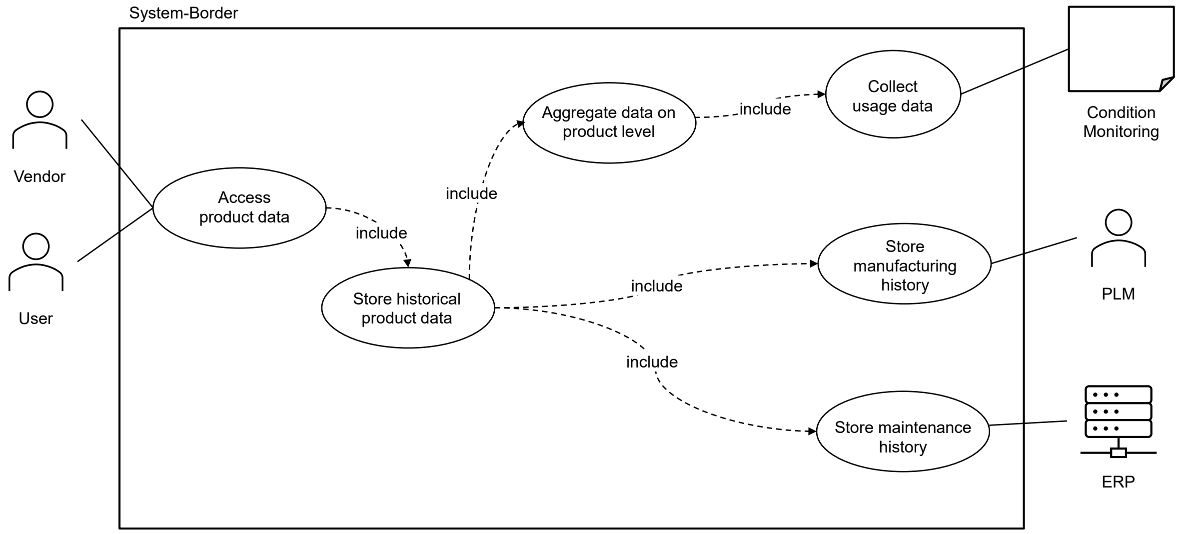

Figure 34 *UML-diagram for use case Create Digital Product Image*

UML-diagram for use case Deliver Consumables/Supplies

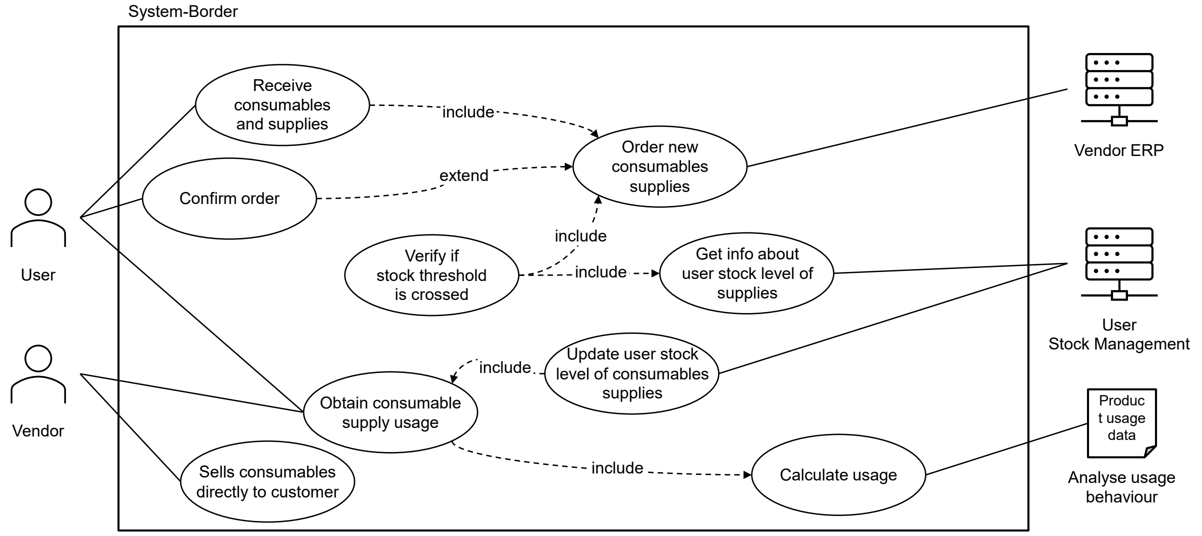

Figure 35 *UML-diagram for use case Deliver Consumables/Supplies*

UML-diagram for use case Derive New Products/Services

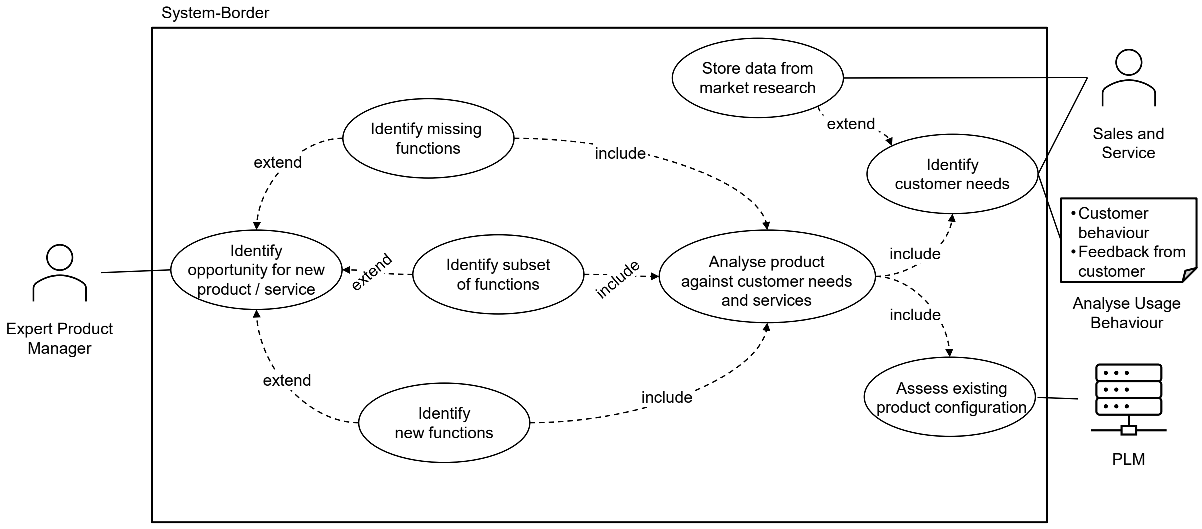

Figure 36 *UML-diagram for use case Derive New Products/Services*

UML-diagram for use case Improve Products/Services

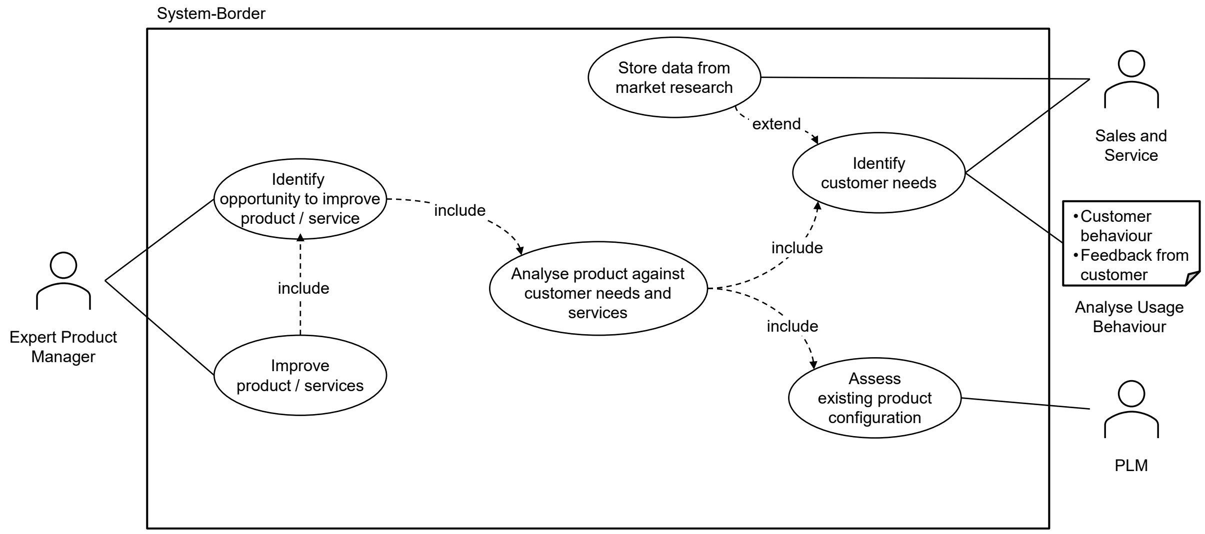

Figure 37 *UML-diagram for use case Improve Products/Services*

UML-diagram for use case Offer Data Analytics

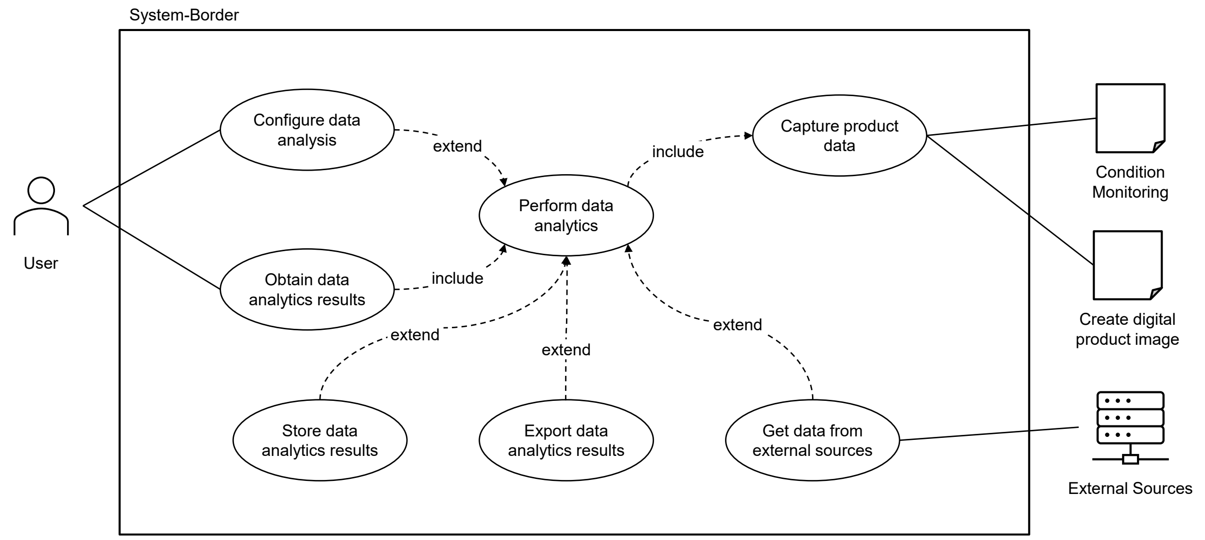

Figure 38 *UML-diagram for use case Offer Data Analytics*

UML-diagram for use case Offer Subscription

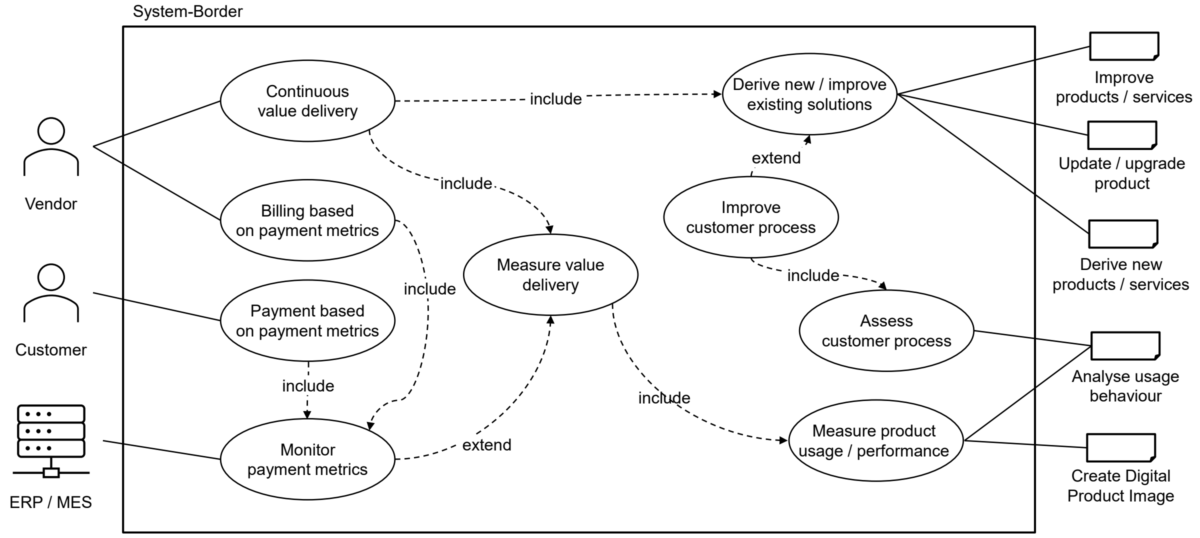

Figure 39 *UML-diagram for use case Offer Subscription*

UML-diagram for use case Optimize Operation

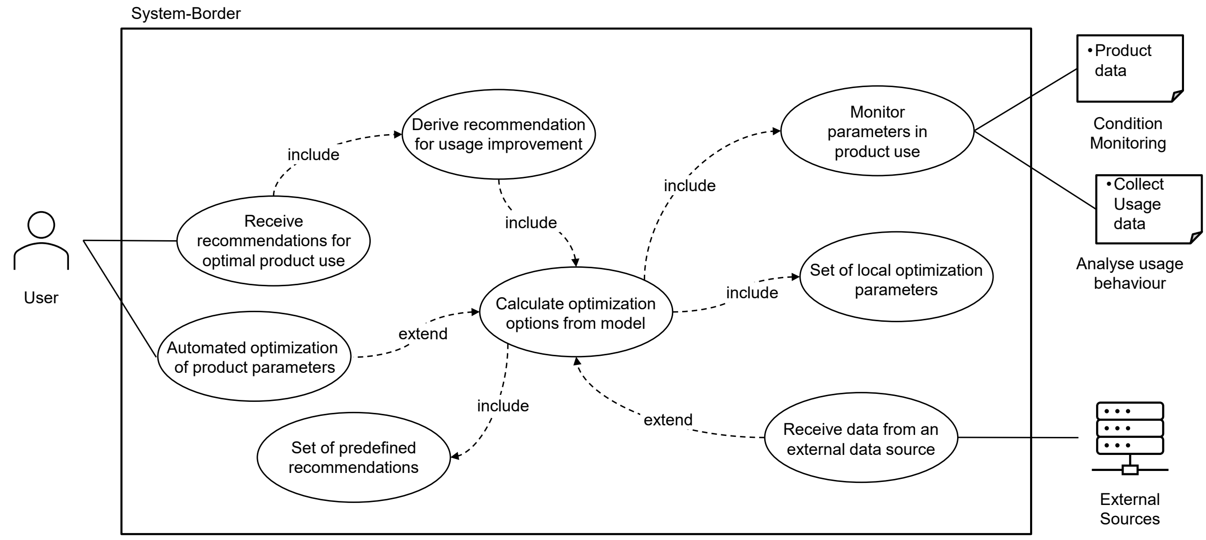

Figure 40 *UML-diagram for use case Optimize Operation*

UML-diagram for use case Provide Predictive Maintenance

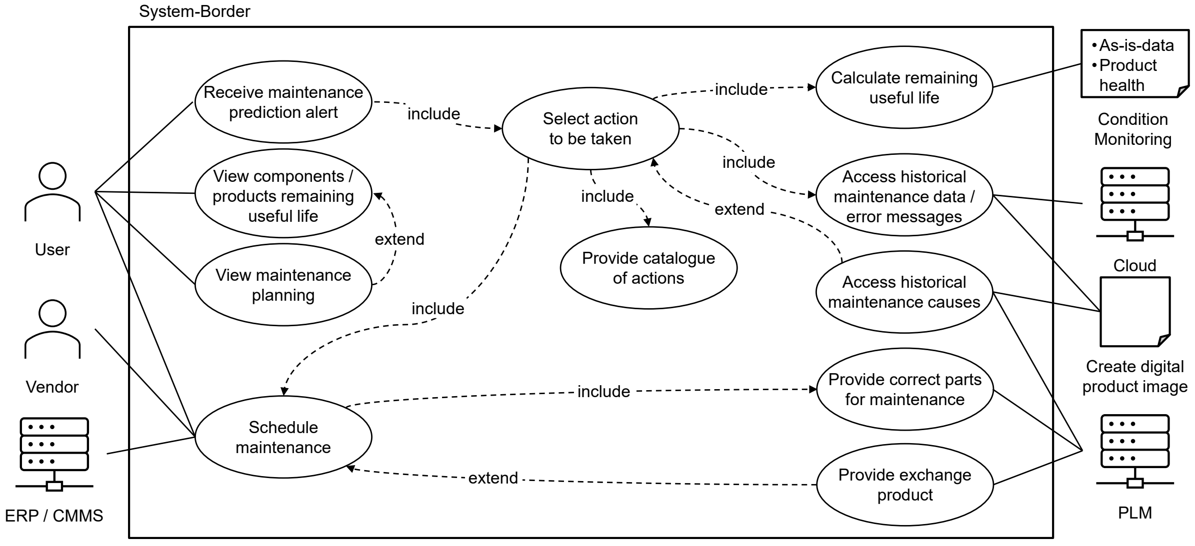

Figure 41 *UML-diagram for use case Provide Predictive Maintenance*

UML-diagram for use case Update Products

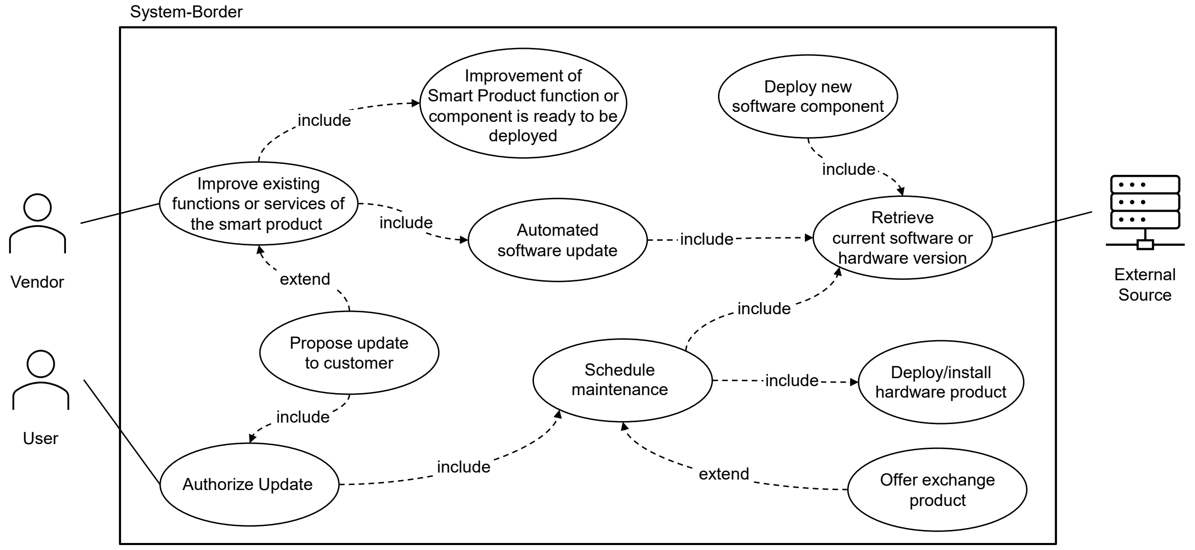

Figure 42 *UML-diagram for use case Update Products*

UML-diagram for use case Upgrade Products

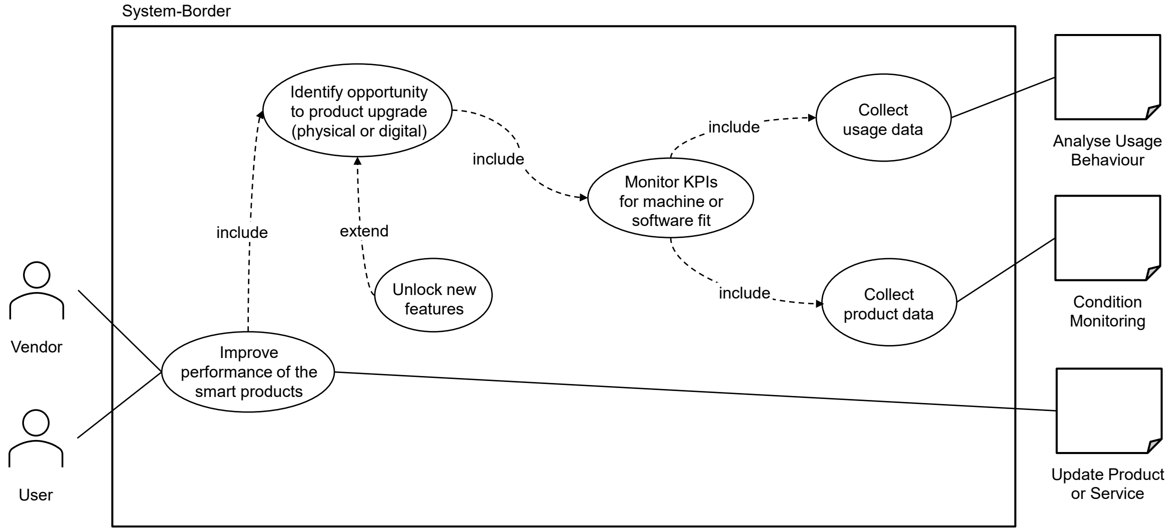

Figure 43 *UML-diagram for use case Upgrade Products*

Analysis of Reference Architectures

Table 9: Overview and results for analyzed architectures

Criteria / Architecture	Expertise Requirements	Modularity Aspects	Availability of Different Views	Smart Product Orientation	Sensor to Cloud Coverage	Security and Identity Coverage	Device Management	Digital Twin Coverage	Data Governance	Scalability
ISO IoT	Requires broad IoT standards knowledge	Highly modular with standardized interfaces	Functional, Information, and Domain views	General IoT focus without specific smart product orientation	Covers entire IoT spectrum from sensor to cloud	Strong focus on security and identity management in IoT	General IoT device management capabilities	Support for digital twin concepts, but not specifically outlined	Data standards and governance for IoT	Designed for scalability across diverse IoT ecosystems
IIRA	Requires industrial IoT knowledge	Modular design for industrial systems	Functional, Information, and Domain views	Industrial IoT with some focus on smart products	Comprehensive from sensor to cloud in industrial contexts	Strong security and identity management in industrial IoT contexts	Focused on industrial IoT device management	Digital twin concepts for industrial applications	Data governance focused on industrial data integrity and sharing	Scalable in industrial IoT settings

RAMI 4.0	Requires comprehensive industrial and IT/OT knowledge	Highly modular with clear interoperability standards	Functional, Life Cycle & Value Stream, Hierarchy Levels Views	Broad industrial focus, indirectly supports smart products	Complete coverage from sensor to cloud including edge computing	Comprehensive security and identity management across IT and OT	Broad coverage, including industrial device management	Detailed digital twin integration across industrial sectors	Comprehensive data governance across lifecycle and hierarchy levels	Scalable across industrial sectors with various technologies
Porter & Heppelman	Requires understanding of smart products and IoT	Modular aspects are not the main focus	Limited to strategic and product lifecycle views	Direct focus on smart, connected products	Not specifically addressed but implied	Security considerations are part of the overall strategy	Not the main focus	Considered within the broader context of smart products	Data governance as part of product and ecosystem strategy	Scalability is implied through product and ecosystem growth
Azure	Requires cloud computing and Azure-specific knowledge	Highly modular, supports a wide range of services	Various architectural views for solution design	General cloud solutions with IoT services	Comprehensive cloud, IoT, and edge solutions	Comprehensive security and identity solutions	Advanced device management capabilities	Comprehensive digital twin capabilities	Comprehensive data governance and compliance solutions	Highly scalable cloud architecture
PTC	Requires knowledge of PTC's platforms and industrial IoT	Modular, with focus on IoT and AR integration	Focused on IoT and AR perspectives	Direct focus on smart product development and management	Comprehensive IoT solutions from sensor to product to cloud	Security and identity management for IoT and AR	IoT device management with focus on lifecycle management	Digital twin integration in IoT and AR	Data governance in IoT and AR contexts	Scalable IoT and AR solutions
AWS	Requires AWS cloud services knowledge	Highly modular, supports a wide range of services	Various architectural views for solution design	General cloud solutions with IoT services	Comprehensive cloud, IoT, and edge solutions	Comprehensive security and identity solutions	Advanced device management capabilities	Comprehensive digital twin capabilities	Comprehensive data governance and compliance solutions	Highly scalable cloud architecture
Siemens Mindsphere	Requires knowledge of Siemens' platform and	Modular, but with some focus on Siemens-	Focus on operational and analytics	Industrial IoT with smart product analytics	Comprehensive industrial IoT solutions from sensor to cloud	Strong security focus, particularly in industrial IoT	Industrial device management with Siemens-	Strong digital twin capabilities for industrial	Data governance with focus on industrial data	Scalable in industrial IoT contexts

	industrial IoT	specific solutions	perspectives				specific tools	applications		
Google Cloud	Requires GCP cloud services knowledge	Highly modular, supports a wide range of services	Focus on computing, networking, and storage views	General cloud solutions with IoT services	Comprehensive cloud, IoT, and edge solutions	Comprehensive security and identity solutions	Advanced device management capabilities	Comprehensive digital twin capabilities	Comprehensive data governance and compliance solutions	Highly scalable cloud architecture
Adamos	Requires knowledge of smart manufacturing ecosystems	Modular with emphasis on manufacturing ecosystem integration	Focused on digital transformation in manufacturing	Direct focus on smart manufacturing and products	Comprehensive from sensor to cloud in manufacturing contexts	Security and identity management in manufacturing contexts	Focused on manufacturing device management	Digital twin concepts in manufacturing contexts	Data governance in manufacturing ecosystems	Scalable in manufacturing contexts
Alibaba Cloud	Requires Alibaba cloud services knowledge	Highly modular, supports a wide range of services	Focus on cloud computing and data processing views	General cloud solutions with IoT services	Comprehensive cloud, IoT, and edge solutions	Comprehensive security and identity solutions	Advanced device management capabilities	Digital twin capabilities for cloud applications	Comprehensive data governance and compliance solutions	Highly scalable cloud architecture
FIWARE	Requires smart city and IoT platform knowledge	Modular, with focus on open standards and interoperability	Focus on context, data models, and interoperability	Smart city and IoT focus, indirectly supports smart products	Comprehensive IoT platform from sensor to cloud	Focus on security and identity in smart solutions	IoT device management with smart city applications	Digital twin concepts with a focus on IoT and smart cities	Data governance with a focus on interoperability	Scalable IoT platform for various applications
IBM Industry 4.0	Requires industrial IoT and IBM's platform knowledge	Modular, with focus on integration of IBM's technologies	Focus on industry-specific solutions and analytics	Industry-specific IoT solutions, indirectly supports smart	Comprehensive industrial IoT from sensor to cloud	Comprehensive security in industrial IoT	Industrial IoT device management	Digital twin integration in industrial IoT	Data governance in industrial IoT contexts	Scalable in industrial IoT settings

				products						
Bosch IoT	Requires knowledge of Bosch's IoT solutions	Highly modular, supports diverse IoT applications	Focus on device management and data processing	Industrial IoT solutions, some smart product features	Comprehensive IoT solutions from sensor to cloud	Strong security and identity management solutions	IoT device management solutions	Digital twin solutions for IoT applications	Data governance for IoT solutions	Scalable IoT solutions across various industries
AVEVA (Wonderware)	Requires industrial software and digital transformation knowledge	Modular, focuses on industrial software solutions	Focus on engineering, operations, and performance management	General industrial software, indirect smart product support	Comprehensive industrial IoT solutions from sensor to cloud	Comprehensive security and identity management in industrial context	Industrial device management with a focus on operations	Digital twin capabilities within industrial software solutions	Data governance and compliance in industrial software solutions	Scalable in industrial applications

Architectural Blueprints for the development of smart products

Blueprints – Functional View

Table 10 *Blueprints - Functional view (UC: Use Case, Nc: Necessity, MH: must have, SH: should have, NTH: nice to have)*

UC	Functional element	Constituting requirement	Nc	Cluster
Analyse Usage Behaviour	Connect to external API	Collect insights from app analytics platform (e.g.: to get web/mobile app usage information)	NTH	API
	Connect to business IT-System	Collect customer segment data	SH	
		Access customer data (e.g. CRM, customers database)	SH	
	Connect to internal platform	Access product condition data (link with condition monitoring use-case)	SH	
	Data clustering	Run automatic pattern analysis on collected usage data	SH	Data analysis
		Create behavior profiles agnostic of customer information - only based on usage data	MH	
	Data preprocessing	Aggregate and/or pre-process usage data	MH	
		Anonymize customer data for analysis and comparison	SH	
	Data exploration	Explore data to derive results from it	MH	
		Segment data to scope analysis to a defined data selection	MH	
	Data ingestion	Collect usage data	MH	Endpoint
		Capture user input	SH	
	Scripting	Provide means/application to further process the collected and analyzed data output	SH	Logic
		Configure the automatic archival or deletion of the collected data	NTH	
	Automation	Execute automation workflow (e.g.: aggregation -> data analysis -> reporting)	NTH	
	Database storage	Store data analysis results	SH	Persistence
		Store behavior profiles	SH	
	File storage	Store automatically generated usage behavior analysis reports	SH	
	Capture relevant data	Capture user input	SH	Smart Product
		Collect usage data	MH	
	Transmit relevant data	Capture user input	SH	
		Collect usage data	MH	
	Forms	Collect user feedback on why something was done in a certain way in the application	NTH	UI
	Low-code/No-code environment	Provide application for working with the collected and analyzed data	MH	
		Provide user interface for product management expert	SH	
	Dashboard	Visualize data analysis results	MH	
Assist Operation	Data clustering	Cluster usage behavior to compare them to predefined clusters	SH	Data analysis
	Data preprocessing	Pre-process data to a joint format / arrangement	SH	
	Data ingestion	Collect usage data	SH	Endpoint
		Collect product configuration	MH	
		Capture user input	NTH	
	Data serving	Provide manuals to the user	SH	
		Provide means to control the smart product	SH	
	Rule engine	Rule engine (tip based on usage)	MH	Logic
		Provide information in the user interface about the intended usage of the product	MH	

UC	Functional element	Constituting requirement	Nc	Cluster
	Database storage	Store user data	NTH	Persistence
		Store history of assistance interventions (to improve individualized assistance)	SH	
	File storage	Provide manuals to the user	SH	
	Capture relevant data	Capture user input	NTH	Smart Product
		Collect usage data	SH	
		Capture product configuration and usage	MH	
	Receive data	Actuate and/or configure product remotely	SH	
	Transmit relevant data	Capture user input	NTH	
		Collect usage data	SH	
		Capture product configuration and usage	MH	
	Actuate on command	Actuate and/or configure product remotely	SH	
	Notification	Provide information in the user interface about the intended usage of the product	MH	UI
		Notify the user when a required measure would need to be taken	MH	
	Forms	Collect feedback from user	NTH	
	Low-code/No-code environment	Pre-process data to a joint format / arrangement	SH	
	Control panel	Capture user input via product control panel provided by vendor	NTH	
	Dashboard	Provide interface to assess usage behavior to adapt operation assistance	NTH	
Condition Monitoring	Connect to external API	Collect environment data from external sources	NTH	API
	Connect to business IT-System	Collect process data from external sources (ERP/MES)	NTH	
	Data preprocessing	Aggregate and/or pre-process product data	MH	Data analysis
		Compute incoming data features	MH	
	Data ingestion	Collect product data	MH	Endpoint
	Rule engine	Perform notifications actions defined by monitoring rule(s) set	SH	Logic
		Evaluate rule(s) set (=>rule engine)	SH	
	Database storage	Provide hot/warm storage of data	MH	Persistence
	Capture relevant data	Collect product data	MH	Smart Product
	Transmit relevant data	Collect product data	MH	
	Notification	Perform notifications actions defined by monitoring rule(s) set	SH	UI
	Low-code/No-code environment	Provide UI to define monitoring rule(s) set	SH	
	Dashboard	Visualize product condition	MH	
Create Dig Product Image	Connect to business IT-System	Access [remote] data sources (PLM / ERP)	SH	API
		Access manufacturing history from PLM	SH	
		Show available spare parts (related to PLM)	NTH	
		Access maintenance history from ERP	NTH	
		Access product configuration from PLM	MH	
	Provide API to serve RPC/data	Provide API enabling access to product image data	NTH	
	Data computation	Compute individual analysis	NTH	Data analysis
	Device metadata collection	Collect product metadata information	SH	Device management

UC	Functional element	Constituting requirement	Nc	Cluster
	Data ingestion	Collect usage and product data	MH	Endpoint
	Database storage	Show error history	SH	Persistence
		Create personalized visualizations for specific customer/role	SH	
		Store historical data	NTH	
		Collect usage and product data	MH	
	Capture relevant data	Collect usage and product data	MH	Smart Product
	Transmit relevant data	Collect usage and product data	MH	
	Low-code/No-code environment	Create personalized analyses for a specific customer/role	NTH	UI
		Create personalized visualizations for specific customer/role	SH	
	Dashboard	Show error history	SH	
		Provide interface to view product data [client & vendor]	MH	
		Visualize product condition	MH	
		Create personalized visualizations for specific customer/role	SH	
Deliver Consumables/Supplies	Connect to business IT-System	Support integration of customer ERP system	SH	API
	Data computation	Compute consumption rate and consumable demand	SH	Data analysis
	Data computation	Compare consumable demand and stock	MH	
	Data ingestion	Collect usage data	MH	Endpoint
		Capture consumables' usage	SH	
	Rule engine	Evaluate quantity and remaining inventory to determine if an order is needed	MH	Logic
	Automation	Recommend the restocking of consumables/supplies automatically (based on usage rate and stock level)	SH	
	Capture relevant data	Capture consumables' usage	SH	Smart Product
		Collect usage data	MH	
	Transmit relevant data	Capture consumables' usage	SH	
		Collect usage data	MH	
		Collect usage data from smart product	SH	
	Notification	Notify user about the automatic order process	MH	UI
	Control panel	Provide UI to confirm consumables/suppliers order	NTH	
		Provide interface for user/vendor to configure re-supply logic	NTH	
	Dashboard	Provide interface for user to monitor consumables usage	SH	
Derive New Products/Services	Connect to external API	Access feedback from customers and resellers (for existing service and existing product)	SH	API
	Connect to business IT-System	Access existing product configuration (e.g. from PLM)	MH	
		Access market research data	SH	
	Connect to internal platform	Access customer behavior profiles (link with Analyze Usage Behavior use-case)	MH	
	Data clustering	Cluster customer needs to compare them to existing functions	SH	Data analysis
		Create behavior profiles agnostic of customer information - only based on usage data	MH	
	Data preprocessing	Pre-process data to a joint format / arrangement	SH	
	Data exploration	Analyze existing product configurations	NTH	
		Analyze customer and reseller feedback (for existing service and existing product)	NTH	
		Explore and analyze market research data	SH	

UC	Functional element	Constituting requirement	Nc	Cluster
		Compare product as-is state to customer needs	MH	
		Provide data exploration platform to identify missing functions	SH	
	Data ingestion	Collect usage data	MH	Endpoint
		Collect product data	MH	
	Automation	Automated creation and updating reports on improving existing functions	NTH	Logic
		Collect and analyze data automatically through a pipeline	NTH	
		Identify missing functions automatically	NTH	
	Database storage	Access feedback from customers and resellers (for existing service and existing product)	SH	Persistence
		Manage identified opportunities (Collect, add description, forward to others)	MH	
	Capture relevant data	Collect usage data	MH	Smart Product
		Collect product data	MH	
	Transmit relevant data	Collect usage data	MH	
		Collect product data	MH	
	Forms	Collect user feedback on why something was done in a certain way in the application	SH	UI
	Low-code/No-code environment	Create UIs and reports addressed to product managers	SH	
		Provide interface to manually analyze data	MH	
		Pre-process data to a joint format / arrangement	SH	
	Dashboard	UI to query and visualize data	MH	
		Access feedback from customers and resellers (for existing service and existing product)	SH	
Improve Products/Services	Connect to external API	Access feedback from customers and resellers (for existing service and existing product)	MH	API
	Connect to business IT-System	Access existing product configuration (e.g. from PLM)	MH	
		Access market research data	SH	
	Connect to internal platform	Access customer behavior profiles (link with Analyze Usage Behavior use-case)	MH	
	Data clustering	Compare product as is situation to derive customer needs	MH	Data analysis
		Cluster customer needs to compare them to existing functions	MH	
		Create behavior profiles agnostic of customer information - only based on usage data	MH	
	Data preprocessing	Pre-process data to a joint format / arrangement	SH	
	Data exploration	Analyze existing product configurations	SH	
		Analyze customer and reseller feedback (for existing service and existing product)	MH	
		Explore and analyze market research data	NTH	
		Compare product as-is state to customer needs	MH	
		Provide data exploration platform to identify missing functions	SH	
	Data ingestion	Collect usage data	MH	Endpoint
		Collect product data	MH	
	Automation	Automated creation and updating reports on improving existing functions	NTH	Logic
		Collect and analyze data automatically through a pipeline	NTH	
		Identify missing functions automatically	NTH	
	Database storage	Access feedback from customers and resellers (for existing service and existing product)	MH	Persistence
		Manage identified opportunities (Collect, add description, forward to others)	MH	

UC	Functional element	Constituting requirement	Nc	Cluster
	Capture relevant data	Collect usage data	MH	Smart Product
		Collect product data	MH	
	Transmit relevant data	Collect usage data	MH	
		Collect product data	MH	
	Forms	Collect user feedback on why something was done in a certain way in the application	SH	UI
	Low-code/No-code environment	Create UIs and reports addressed to product managers	SH	
		Provide interface to manually analyze data	MH	
		Pre-process data to a joint format / arrangement	SH	
	Dashboard	UI to query and visualize data	MH	
		Access feedback from customers and resellers (for existing service and existing product)	MH	
Offer Data Analytics	Connect to external API	Collect data from external sources	SH	API
	Provide API to serve RPC/data	Provide API enabling analysis output export	SH	
	Data exploration	Provide means to use the product's data analysis results and extend them	MH	Data analysis
		Segment data to scope analysis to a defined data selection	MH	
	Data ingestion	Collect usage data	NTH	Endpoint
		Collect product data	MH	Endpoint
	Data serving	Export data analysis output	SH	Endpoint
	Scripting	Configure data analysis (computation logic - KPI)	MH	Logic
	Automation	Execute data analysis workflows	SH	
		Configure data analysis workflows (automation/execution frequency)	SH	
	Database storage	Store data analysis results	SH	Persistence
		Create data selections (query/view to be used)	MH	
	Capture relevant data	Collect usage data	NTH	Smart Product
		Collect product data	MH	
	Transmit relevant data	Collect usage data	NTH	
		Collect product data	MH	
	Dashboard	Visualize data analysis results	MH	UI
Offer Subscription	Connect to external API	Collect necessary data to compute payment metrics from external systems	MH	API
		Support integration of customer ERP system	NTH	
		Enable billing based on payment metrics through ERP	SH	
	Connect to business IT-System	Connect to process management system to see customer process	NTH	
		Integrate internal IT-System (ERP)	MH	
		Access ERP customer data for billing	MH	
	Connect to internal platform	Access product or service usage data	MH	
		Collect necessary data to compute payment metrics from internal systems	NTH	
	Data computation	Compute metrics linked to payment criteria	MH	Data analysis
		Compute product or service value	MH	
	Data ingestion	Access product or service usage data	MH	Endpoint
		Collect necessary data to compute payment metrics from products	MH	
		Collect necessary data to compute value delivery	SH	
	Rule engine	Trigger billing, payment, etc. operations	MH	Logic
	Automation	Execute billing process automatically	SH	

UC	Functional element	Constituting requirement	Nc	Cluster
	Capture relevant data	Collect necessary data to compute value delivery	SH	Smart Product
	Transmit relevant data	Collect necessary data to compute value delivery	SH	
	Notification	Receive notifications about ongoing billing process, new bills etc.	MH	UI
	Forms	Gather feedback about the measurement of the value delivered by the service	NTH	
	Control panel	Configure key payment figures (when not handled through the ERP)	MH	
	Dashboard	Capture analysis and decision making process for transparency [vendor-side functionality]	SH	
		Provide an interface to access relevant product data and billing	NTH	
		Visualize value delivery and payment	SH	
Optimize Operation	Connect to external API	Enrich data with external data sources (e.g. product quality from ERP)	NTH	API
	Connect to internal platform	Aggregate usage and product data	MH	
	Data clustering	Profile usage data to map it to a set of optimization parameters	NTH	Data analysis
	Data preprocessing	Aggregate usage and product data	MH	
	Data exploration	Analyze data to derive new predefined measures/recommendations [vendor]	MH	
	Data computation	Compute optimization options applying the analytical optimization model	MH	
		Calculate optimization options from model (heuristic model)	MH	
		Automatically process data to derive predefined measures/recommendations	NTH	
	Data ingestion	Collect usage data	MH	Endpoint
		Collect product data	MH	
	Data serving	Apply optimized settings to the product	SH	Endpoint
	Scripting	Configure optimization model for data analysis (e.g. parameters, analytical function, AI)	MH	Logic
	Automation	Apply optimization model	MH	
	Database storage	Store pre-defined recommendations	MH	Persistence
	Capture relevant data	Collect usage data	MH	Smart Product
		Collect product data	MH	
	Receive data	Apply optimized settings to the product	SH	
	Transmit relevant data	Collect usage data	MH	
		Collect product data	MH	
	Actuate on command	Apply optimized settings to the product	SH	
	Notification	Notify the user when a required measure would need to be taken	MH	UI
	Dashboard	Collect feedback about model performance [labeling]	NTH	
Provide Predictive Maintenance	Connect to business IT-System	Coordinate maintenance scheduling with parties / systems	NTH	API
		Access manufacturing history from PLM	NTH	
		Access maintenance history from ERP	NTH	
	Connect to internal platform	Access historical maintenance data	SH	
	ML model training	Train the ML model by feeding data and labeling results	NTH	Data analysis

UC	Functional element	Constituting requirement	Nc	Cluster
	Data preprocessing	Aggregate and/or pre-process product data	SH	
		Compute incoming data features	SH	
	Data computation	Compute RUL AI-based	NTH	
		Compute product or components Remaining Useful Life (RUL)	MH	
	Data ingestion	Collect product data	MH	Endpoint
	Rule engine	Take actions from a catalog based on model output	MH	Logic
	Automation	Coordinate maintenance scheduling with parties / systems	NTH	
	Database storage	Show error history	NTH	Persistence
		Store historical data	NTH	
	Capture relevant data	Collect product data	MH	Smart Product
	Transmit relevant data	Collect product data	MH	
	Notification	Notify the different parties about predicted maintenance events	MH	UI
	Low-code/No-code environment	Provide UI to configure catalogue of actions	SH	
	Dashboard	Show error history	NTH	
		Provide UI to view product's Remaining Useful Life (RUL)	MH	
		Provide UI to view maintenance planning	MH	
Update Products	Connect to business IT-System	Obtain software version from physical product or edge device	MH	API
	Remote update	Update device software Over-The-Air (OTA)	MH	Device management
		Schedule device software updates (OTA update manager service)	MH	
	Device metadata collection	Obtain overview of software versions on smart products	SH	
	Data ingestion	Obtain hardware deployed version from the physical product	MH	Endpoint
		Obtain software version from physical product or edge device	MH	
	Data serving	Update device software Over-The-Air (OTA)	MH	
		Provide robust fallback solution (e.g. permalink to revert software/firmware)	MH	
	Rule engine	Provide A-B testing capabilities	NTH	Logic
	Scripting	Schedule device software updates (OTA update manager service)	MH	
	Automation	Update device software automatically (push)	SH	
	File storage	Keep key firmware versions to enable incremental updates	MH	Persistence
		Provide data integrity and verification hashed (source/bins)	MH	
		Store software updates in file repository	MH	
	Receive data	Update device software Over-The-Air (OTA)	MH	Smart Product
		Provide self-updating capabilities	SH	
	Transmit relevant data	Obtain hardware deployed version from the physical product	MH	
		Provide self-updating capabilities	SH	
		Obtain software version from physical product or edge device	MH	

UC	Functional element	Constituting requirement	Nc	Cluster
	Notification	Receive notifications during update process (update available, update successfully deployed, update error)	MH	UI
	Control panel	Provide A-B testing capabilities	NTH	
		Provide frontend to the update center to the customer	SH	
		Notify customer that an update/upgrade can be deployed	MH	
		Provide interface to schedule a maintenance slot to the product's end-user	MH	
		Provide interface to validate and/or authorize the update/upgrade's deployment	MH	
	Dashboard	Obtain overview of software versions on smart products	SH	
Upgrade Products	Connect to business IT-System	Obtain software version from physical product or edge device	MH	API
		Manage or exchange products for upgrades	NTH	
	Connect to internal platform	Verify license validity prior to upgrade (vendor side)	MH	
	Data exploration	Evaluate product KPI to derive opportunities for upgrade	MH	Data analysis
	Remote update	Update device software Over-The-Air (OTA)	MH	Device management
		Schedule device software updates (OTA update manager service)	SH	
	Data ingestion	Collect usage data	SH	Endpoint
		Obtain hardware deployed version from the physical product	SH	
		Obtain software version from physical product or edge device	MH	
		Collect product data	MH	
	Data serving	Update device software Over-The-Air (OTA)	MH	
		Control features availability on product (locking/unlocking of features)	SH	
		Provide robust fallback solution (e.g. permalink to revert software/firmware)	MH	
	Rule engine	Manage or exchange products for upgrades	NTH	Logic
		Provide A-B testing capabilities	NTH	
	Scripting	Schedule device software updates (OTA update manager service)	SH	
	File storage	Keep key firmware versions to enable incremental updates	MH	Persistence
		Provide data integrity and verification hashed (source/bins)	MH	
		Store software updates in file repository	MH	
	Capture relevant data	Collect usage data	SH	Smart Product
		Collect product data	MH	
	Receive data	Control features availability on product (locking/unlocking of features)	SH	
	Receive data	Update device software Over-The-Air (OTA)	MH	
	Transmit relevant data	Collect usage data	SH	
		Obtain hardware deployed version from the physical product	SH	
		Obtain software version from physical product or edge device	MH	
		Collect product data	MH	
	Notification	Receive notifications during update process (update available, update successfully deployed, update error)	MH	UI
	Control panel	Provide A-B testing capabilities	NTH	
		Provide frontend to the update center to the customer	MH	

UC	Functional element	Constituting requirement	Nc	Cluster
		Notify customer that an update/upgrade can be deployed	MH	
		Provide interface to schedule a maintenance slot to the product's end-user	SH	
		Provide interface to validate and/or authorize the update/upgrade's deployment	MH	
	Dashboard	Inform customer about upgrade opportunity	SH	

Blueprints – Task View

Table 11 *Blueprints - Task view (UC: Use Case, S: Processual Stage, DT: Data transformation)*

UC	Task element	Constituting requirement	Domain	S
Analyze Usage Behaviour	Transmit relevant data	Collect usage data	Node	Data gathering
	Capture relevant data	Collect usage data		
	Data ingestion	Collect usage data	Edge	
		Capture user input		
	Data ingestion	Collect usage data	Backend	
		Capture user input		
	Connect to external API	Collect insights from app analytics platform (e.g. to get web/mobile app usage information)		
	Connect to business IT-System	Access customer data (e.g. CRM, customers database)		
		Collect customer segment data		
	Connect to internal platform	Access product condition data (link with condition monitoring use-case)		
	Forms	Collect user feedback on why something was done in a certain way in the application		
	Data preprocessing	Anonymize customer data for analysis and comparison	Node	Data transformation
		Aggregate and/or pre-process usage data		
	Automation	Execute automation workflow (e.g.: aggregation -> data analysis -> reporting)		
	Scripting	Configure the automatic archival or deletion of the collected data		
		Provide means/application to further process the collected and analyzed data output		
	Data clustering	Run automatic pattern analysis on collected usage data	Edge	
		Create behavior profiles agnostic of customer information - only based on usage data		
	Data preprocessing	Anonymize customer data for analysis and comparison		
		Aggregate and/or pre-process usage data		

UC	Task element	Constituting requirement	Domain	S
	Automation	Execute automation workflow (e.g.: aggregation -> data analysis -> reporting)		
	Scripting	Configure the automatic archival or deletion of the collected data		
		Provide means/application to further process the collected and analyzed data output		
	Data exploration	Segment data to scope analysis to a defined data selection	Backend	
	Low-code/No-code environment	Provide user interface for product management expert		
		Provide application for working with the collected and analyzed data		
	Dashboard	Visualize data analysis results		
	File storage	Store automatically generated usage behavior analysis reports		
	Database storage	Store data analysis results		
		Store behavior profiles		
	Data clustering	Run automatic pattern analysis on collected usage data		
		Create behavior profiles agnostic of customer information - only based on usage data		
	Data exploration	Explore data to derive results from it		
	Data preprocessing	Anonymize customer data for analysis and comparison		
		Aggregate and/or pre-process usage data		
	Automation	Execute automation workflow (e.g.: aggregation -> data analysis -> reporting)		
	Scripting	Configure the automatic archival or deletion of the collected data		
		Provide means/application to further process the collected and analyzed data output		
	Dashboard	Visualize data analysis results	Backend	Action
	File storage	Store automatically generated usage behavior analysis reports		
	Database storage	Store data analysis results		
		Store behavior profiles		
	Data exploration	Explore data to derive results from it		
Assist Operation	Capture relevant data	Capture user input	Node	Data gathering
		Capture product configuration and usage		
	Transmit relevant data	Capture user input		
		Capture product configuration and usage		
	Control panel	Capture user input via product control panel provided by vendor	Edge	

UC	Task element	Constituting requirement	Domain	S
	Data ingestion	Collect usage data		
		Collect product configuration		
	Forms	Collect feedback from user	Backend	
	Database storage	Store user data		
	Control panel	Capture user input via product control panel provided by vendor		
	Data ingestion	Collect usage data		
		Collect product configuration		
	Data preprocessing	Pre-process data to a joint format / arrangement	Edge	Data transformation
	Low-code/No-code environment	Pre-process data to a joint format / arrangement	Backend	
	Data preprocessing	Pre-process data to a joint format / arrangement		
	Data clustering	Cluster usage behavior to compare them to predefined clusters		
	Rule engine	Provide information in the user interface about the intended usage of the product		
		Rule engine (tip based on usage)		
	Dashboard	Provide interface to assess usage behavior to adapt operation assistance		
	Receive data	Actuate and/or configure product remotely	Node	Action
	Actuate on command	Actuate and/or configure product remotely		
	File storage	Provide manuals to the user	Backend	
	Database storage	Store history of assistance interventions (to improve individualized assistance)		
	Notification	Provide information in the user interface about the intended usage of the product		
		Notify the user when a required measure would need to be taken		
	Data serving	Provide manuals to the user		
		Provide means to control the smart product		
Condition Monitoring	Transmit relevant data	Collect product data	Node	Data gathering
	Capture relevant data	Collect product data		
	Data ingestion	Collect product data	Edge	
	Database storage	Provide hot/warm storage of data		
	Data ingestion	Collect product data	Backend	
	Low-code/No-code environment	Provide UI to define monitoring rule(s) set		

UC	Task element	Constituting requirement	Domain	S
	Connect to business IT-System	Collect process data from external sources (ERP/MES)		
	Connect to external API	Collect environment data from external sources		
	Database storage	Provide hot/warm storage of data		
	Data preprocessing	Compute incoming data features	Node	Data transformation
		Aggregate and/or pre-process product data		
	Data preprocessing	Compute incoming data features	Edge	
		Compute incoming data features		
		Aggregate and/or pre-process product data		
	Rule engine	Perform notifications actions defined by monitoring rule(s) set	Backend	
		Evaluate rule(s) set (=>rule engine)		
	Data preprocessing	Compute incoming data features		
		Compute incoming data features		
		Aggregate and/or pre-process product data		
	Notification	Perform notifications actions defined by monitoring rule(s) set	Edge	Action
	Rule engine	Perform notifications actions defined by monitoring rule(s) set	Backend	
		Evaluate rule(s) set (=>rule engine)		
	Notification	Perform notifications actions defined by monitoring rule(s) set		
	Dashboard	Visualize product condition		
Create Digital Product Image	Transmit relevant data	Collect usage and product data	Node	Data gathering
	Capture relevant data	Collect usage and product data		
	Data ingestion	Collect usage and product data	Edge	
	Connect to business IT-System	Access manufacturing history from PLM	Backend	
		Show available spare parts (related to PLM)		
		Access maintenance history from ERP		
		Access product configuration from PLM		
		Access [remote] data sources (PLM / ERP)		
	Database storage	Show error history		
		Store historical data		
	Device metadata collection	Collect product metadata information		

UC	Task element	Constituting requirement	Domain	S
	Low-code/No-code environment	Create personalized visualizations for specific customer/role		
	Data ingestion	Collect usage and product data		
	Dashboard	Visualize product condition	Backend	Data transformation
		Show error history		
		Provide interface to view product data [client & vendor]		
	Low-code/No-code environment	Create personalized analyses for a specific customer/role		
	Data computation	Compute individual analysis		
	Dashboard	Visualize product condition	Backend	Action
		Show error history		
		Provide interface to view product data [client & vendor]		
	Provide API to serve RPC/data	Provide API enabling access to product image data		
Deliver Consumables/Supplies	Transmit relevant data	Collect usage data	Node	Data gathering
		Collect usage data from smart product		
	Capture relevant data	Capture consumables' usage		
		Collect usage data		
	Data ingestion	Collect usage data	Edge	
		Capture consumables' usage		
	Data ingestion	Collect usage data	Backend	
	Connect to business IT-System	Support integration of customer ERP system		
	Data ingestion	Capture consumables' usage		
	Control panel	Provide interface for user/vendor to configure re-supply logic		
	Rule engine	Evaluate quantity and remaining inventory to determine if an order is needed	Backend	DT
	Data computation	Compare consumable demand and stock		
		Compute consumption rate and consumable demand		
	Automation	Recommend the restocking of consumables/supplies automatically (based on usage rate and stock level)	Node	Action
	Automation	Recommend the restocking of consumables/supplies automatically (based on usage rate and stock level)	Edge	
	Control panel	Provide UI to confirm consumables/suppliers order	Backend	

UC	Task element	Constituting requirement	Domain	S
	Dashboard	Provide interface for user to monitor consumables usage		
	Notification	Notify user about the automatic order process		
	Automation	Recommend the restocking of consumables/supplies automatically (based on usage rate and stock level)		
Derive New Products/Services	Transmit relevant data	Collect usage data	Node	Data gathering
	Capture relevant data	Collect usage data		
		Collect product data		
	Data ingestion	Collect usage data	Edge	
		Collect product data		
	Connect to internal platform	Access customer behavior profiles (link with Analyze Usage Behavior use-case)	Backend	
	Data ingestion	Collect usage data		
		Collect product data		
	Forms	Collect user feedback on why something was done in a certain way in the application		
	Database storage	Access feedback from customers and resellers (for existing service and existing product)		
	Connect to external API	Access feedback from customers and resellers (for existing service and existing product)		
	Connect to business IT-System	Access market research data		
		Access existing product configuration (e.g. from PLM)		
	Data preprocessing	Pre-process data to a joint format / arrangement	Node	Data transformation
	Automation	Identify missing functions automatically		
		Automated creation and updating reports on improving existing functions		
		Collect and analyses data automatically through a pipeline		
	Data preprocessing	Pre-process data to a joint format / arrangement	Edge	
	Automation	Identify missing functions automatically		
		Automated creation and updating reports on improving existing functions		
		Collect and analyses data automatically through a pipeline		
	Low-code/No-code environment	Pre-process data to a joint format / arrangement	Backend	
		Create UIs and reports addressed to product managers		
		Provide interface to manually analyses data		

UC	Task element	Constituting requirement	Domain	S
	Data clustering	Cluster customer needs to compare them to existing functions		
		Create behavior profiles agnostic of customer information - only based on usage data		
	Data preprocessing	Pre-process data to a joint format / arrangement		
	Automation	Identify missing functions automatically		
		Automated creation and updating reports on improving existing functions		
		Collect and analyze data automatically through a pipeline		
	Dashboard	Access feedback from customers and resellers (for existing service and existing product)	Backend	Action
		UI to query and visualize data		
	Data exploration	Provide data exploration platform to identify missing functions		
		Analyze existing product configurations		
		Explore and analyses market research data		
		Compare product as-is state to customer needs		
		Analyze customer and reseller feedback (for existing service and existing product)		
	Database storage	Manage identified opportunities (Collect, add description, forward to others)		
Improve Products/Services	Capture relevant data	Collect usage data	Node	Data gathering
		Collect product data		
	Transmit relevant data	Collect product data		
	Data ingestion	Collect usage data	Edge	
		Collect product data		
	Connect to external API	Access feedback from customers and resellers (for existing service and existing product)	Backend	
	Forms	Collect user feedback on why something was done in a certain way in the application		
	Database storage	Access feedback from customers and resellers (for existing service and existing product)		
	Connect to business IT-System	Access existing product configuration (e.g. from PLM)		
		Access market research data		
	Connect to internal platform	Access customer behavior profiles (link with Analyze Usage Behavior use-case)		
	Data ingestion	Collect usage data		

UC	Task element	Constituting requirement	Domain	S
		Collect product data		
	Data preprocessing	Pre-process data to a joint format / arrangement	Node	DT
	Data preprocessing	Pre-process data to a joint format / arrangement	Edge	
	Data clustering	Cluster customer needs to compare them to existing functions	Backend	
		Compare product as is situation to derive customer needs		
	Data clustering	Create behavior profiles agnostic of customer information - only based on usage data		
	Automation	Identify missing functions automatically		
		Collect and analyze data automatically through a pipeline		
		Automated creation and updating reports on improving existing functions		
	Data preprocessing	Pre-process data to a joint format / arrangement		
	Low-code/No-code environment	Pre-process data to a joint format / arrangement		
	Low-code/No-code environment	Create UIs and reports addressed to product managers		
	Low-code/No-code environment	Provide interface to manually analyze data		
	Database storage	Manage identified opportunities (Collect, add description, forward to others)	Backend	Action
	Dashboard	Access feedback from customers and resellers (for existing service and existing product)		
		UI to query and visualize data		
	Data exploration	Provide data exploration platform to identify missing functions		
		Analyze customer and reseller feedback (for existing service and existing product)		
		Compare product as-is state to customer needs		
		Explore and analyze market research data		
		Analyze existing product configurations		
Offer Data Analytics	Transmit relevant data	Collect usage data	Node	Data gathering
		Collect product data		
	Capture relevant data	Collect usage data		
		Collect product data		
	Connect to external API	Collect data from external sources	Edge	

UC	Task element	Constituting requirement	Domain	S
	Data ingestion	Collect usage data		
	Data ingestion	Collect product data		
	Connect to external API	Collect data from external sources	Backend	
	Data ingestion	Collect usage data		
		Collect product data		
	Scripting	Configure data analysis (computation logic - KPI)		
	Automation	Execute data analysis workflows	Node	Data transformation
		Configure data analysis workflows (automation/execution frequency)		
	Automation	Execute data analysis workflows	Edge	
	Automation	Configure data analysis workflows (automation/execution frequency)		
	Data exploration	Segment data to scope analysis to a defined data selection	Backend	
	Database storage	Create data selections (query/view to be used)		
		Store data analysis results		
	Data exploration	Provide means to use the product's data analysis results and extend them		
	Scripting	Configure data analysis (computation logic - KPI)		
	Automation	Execute data analysis workflows		
		Configure data analysis workflows (automation/execution frequency)		
	Dashboard	Visualize data analysis results	Backend	Action
	Data serving	Export data analysis output		
	Provide API to serve RPC/data	Provide API enabling analysis output export		
Offer Subscription	Transmit relevant data	Collect necessary data to compute value delivery	Node	Data gathering
	Capture relevant data	Collect necessary data to compute value delivery		
	Data ingestion	Access product or service usage data	Edge	
	Data ingestion	Collect necessary data to compute payment metrics from products		
	Data ingestion	Access product or service usage data	Backend	
		Collect necessary data to compute payment metrics from products		
		Access product or service usage data		

UC	Task element	Constituting requirement	Domain	S
	Connect to internal platform	Collect necessary data to compute payment metrics from internal systems		
	Control panel	Configure key payment figures (when not handled through the ERP)		
	Forms	Gather feedback about the measurement of the value delivered by the service		
	Connect to business IT-System	Connect to process management system to see customer process		
		Integrate internal IT-System (ERP)		
	Connect to external API	Collect necessary data to compute payment metrics from external systems		
	Automation	Execute billing process automatically	Backend	DT
	Data computation	Compute metrics linked to payment criteria		
		Compute product or service value		
	Rule engine	Trigger billing, payment, etc. operations		
	Connect to business IT-System	Access ERP customer data for billing	Backend	Action
	Automation	Execute billing process automatically		
	Notification	Receive notifications about ongoing billing process, new bills etc.		
	Connect to external API	Enable billing based on payment metrics through ERP		
		Support integration of customer ERP system		
	Dashboard	Visualize value delivery and payment		
		Capture analysis and decision-making process for transparency [vendor-side functionality]		
		Provide an interface to access relevant product data and billing		
Optimize Operation	Capture relevant data	Collect usage data	Node	Data gathering
		Collect product data		
	Transmit relevant data	Collect usage data		
		Collect product data		
	Data ingestion	Collect usage data	Edge	
		Collect product data		
	Connect to internal platform	Aggregate usage and product data	Backend	
	Connect to external API	Enrich data with external data sources (e.g. product quality from ERP)		

UC	Task element	Constituting requirement	Domain	S
	Data ingestion	Collect usage data		
		Collect product data		
	Data preprocessing	Aggregate usage and product data	Node	Data transformation
	Data preprocessing	Aggregate usage and product data	Edge	
	Data clustering	Profile usage data to map it to a set of optimization parameters	Backend	
	Data exploration	Analyze data to derive new predefined measures/recommendations [vendor]		
	Data preprocessing	Aggregate usage and product data		
	Dashboard	Collect feedback about model performance [labeling]		
	Data computation	Compute optimization options applying the analytical optimization model		
		Calculate optimization options from model (heuristic model)		
		Automatically process data to derive predefined measures/recommendations		
	Scripting	Configure optimization model for data analysis (e.g. parameters, analytical function, AI)		
	Automation	Apply optimization model		
	Receive data	Apply optimized settings to the product	Node	Action
	Actuate on command	Apply optimized settings to the product		
	Data serving	Apply optimized settings to the product	Backend	
	Database storage	Store pre-defined recommendations		
	Notification	Notify the user when a required measure would need to be taken		
Provide Predictive Maintenance	Capture relevant data	Collect product data	Node	Data gathering
	Transmit relevant data	Collect product data		
	Data ingestion	Collect product data	Edge	
	Dashboard	Show error history	Backend	
	Connect to internal platform	Access historical maintenance data		
	Connect to business IT-System	Access maintenance history from ERP		
		Access manufacturing history from PLM		
	Database storage	Show error history		
		Store historical data		
	Low-code/No-code environment	Provide UI to configure catalogue of actions		

UC	Task element	Constituting requirement	Domain	S
	Data ingestion	Collect product data		
	Data preprocessing	Compute incoming data features	Node	Data transformation
		Aggregate and/or pre-process product data		
		Compute incoming data features		
	Data preprocessing	Compute incoming data features	Edge	
		Aggregate and/or pre-process product data		
		Compute incoming data features		
	Data preprocessing	Compute incoming data features	Backend	
	ML model training	Train the ML model by feeding data and labeling results		
	Data computation	Compute product or components Remaining Useful Life (RUL)		
	Data preprocessing	Aggregate and/or pre-process product data		
		Compute incoming data features		
	Rule engine	Take actions from a catalog based on model output		
	Notification	Notify the different parties about predicted maintenance events	Edge	Action
	Connect to business IT-System	Coordinate maintenance scheduling with parties / systems	Backend	
	Automation	Coordinate maintenance scheduling with parties / systems		
	Notification	Notify the different parties about predicted maintenance events		
	Dashboard	Provide UI to view maintenance planning		
		Provide UI to view product's Remaining Useful Life (RUL)		
Update Products	Transmit relevant data	Provide self-updating capabilities	Node	Data gathering
		Obtain software version from physical product or edge device		
		Obtain hardware deployed version from the physical product		
	Data ingestion	Obtain hardware deployed version from the physical product	Edge	
		Obtain software version from physical product or edge device		
		Obtain hardware deployed version from the physical product		
		Obtain software version from physical product or edge device		
	Connect to business IT-System	Obtain software version from physical product or edge device		
	Data ingestion	Obtain hardware deployed version from the physical product	Backend	
		Obtain software version from physical product or edge device		
	Transmit relevant data	Obtain hardware deployed version from the physical product		

UC	Task element	Constituting requirement	Domain	S
		Obtain software version from physical product or edge device		
	Connect to business IT-System	Obtain software version from physical product or edge device		
	Data ingestion	Obtain hardware deployed version from the physical product		
		Obtain software version from physical product or edge device		
	Device metadata collection	Obtain overview of software versions on smart products		
	Control panel	Provide A-B testing capabilities		
		Notify customer that an update/upgrade can be deployed		
		Provide interface to schedule a maintenance slot to the product's end-user		
		Provide interface to validate and/or authorize the update/upgrade's deployment		
		Provide frontend to the update center to the customer		
	Rule engine	Provide A-B testing capabilities	Backend	DT
	Automation	Update device software automatically (push)		
	Scripting	Schedule device software updates (OTA update manager service)		
	Receive data	Provide self-updating capabilities	Node	Action
		Update device software Over-The-Air (OTA)		
	File storage	Keep key firmware versions to enable incremental updates	Backend	
		Provide data integrity and verification hashed (source/bins)		
		Store software updates in file repository		
	Data serving	Update device software Over-The-Air (OTA)		
		Provide robust fallback solution (e.g. permalink to revert software/firmware)		
	Dashboard	Obtain overview of software versions on smart products		
	Automation	Update device software automatically (push)		
	Remote update	Schedule device software updates (OTA update manager service)		
		Update device software Over-The-Air (OTA)		
	Notification	Receive notifications during update process (update available, update successfully deployed, update error)		

UC	Task element	Constituting requirement	Domain	S
Upgrade Products	Transmit relevant data	Collect usage data	Node	Data gathering
		Collect product data		
		Obtain hardware deployed version from the physical product		
		Obtain software version from physical product or edge device		
	Capture relevant data	Collect usage data		
		Collect product data		
	Scripting	Schedule device software updates (OTA update manager service)	Backend	
	Connect to internal platform	Verify license validity prior to upgrade (vendor side)		
	Control panel	Provide A-B testing capabilities		
		Provide interface to schedule a maintenance slot to the product's end-user		
		Provide interface to validate and/or authorize the update/upgrade's deployment		
		Notify customer that an update/upgrade can be deployed		
		Provide frontend to the update center to the customer		
	Data ingestion	Collect usage data		
		Obtain software version from physical product or edge device		
		Obtain hardware deployed version from the physical product		
		Collect product data		
	Rule engine	Manage or exchange products for upgrades	Backend	DT
	Data exploration	Evaluate product KPI to derive opportunities for upgrade		
	Receive data	Control features availability on product (locking/unlocking of features)	Node	Action
		Control features availability on product (locking/unlocking of features)		
		Control features availability on product (locking/unlocking of features)		
		Control features availability on product (locking/unlocking of features)		
		Update device software Over-The-Air (OTA)		
		Update device software Over-The-Air (OTA)		
		Update device software Over-The-Air (OTA)		
		Update device software Over-The-Air (OTA)		

UC	Task element	Constituting requirement	Domain	S
	Data serving	Control features availability on product (locking/unlocking of features)	Backend	
		Provide robust fallback solution (e.g. permalink to revert software/firmware)		
		Update device software Over-The-Air (OTA)		
	Connect to business IT-System	Manage or exchange products for upgrades		
	Dashboard	Inform customer about upgrade opportunity		
	Notification	Receive notifications during update process (update available, update successfully deployed, update error)		
	File storage	Keep key firmware versions to enable incremental updates		
		Store software updates in file repository		
		Provide data integrity and verification hashed (source/bins)		
	Remote update	Schedule device software updates (OTA update manager service)		
		Update device software Over-The-Air (OTA)		

Blueprints – Data View

Table 12 *Blueprints - Data view (UC: Use Case)*

UC	Data element	Constituting task	Cluster	Type
Analyse Usage Behaviour	Customer data	Connect to business IT-System	Master Data	Customer
	Relevant data (individual) from Smart Product	Transmit relevant data	Live	Process
	Stored product data	Database storage	History	Device
	Cloud Analytics	Connect to external API	Field	Global Environment
	Other analyses (e.g. product condition)	Connect to internal platform	Lateral	Asset
	Derived measures	Data clustering	Information	Result
	Aggregated data translated into a format suited for analysis	Data preprocessing		
	Resulting customers clusters	Data clustering	Learning	
Assist Operation	User feedback	Forms	Feedback	Local Environment
	State of product's surroundings	Capture relevant data	Field	
	Process parameters	Capture relevant data	Input	Process
	Measurements during operation	Capture relevant data	Live	
	Quality assurance data (e.g. diagnostics data, logs)	Capture relevant data	Output	
	Thresholds / rules to be used for comparison	Rule engine	Knowledge	Asset
	Manual for smart product	File storage	Product	

UC	Data element	Constituting task	Cluster	Type
	Individual usage advice	Rule engine	Action	Result
Condition Monitoring	State of product's surroundings	Capture relevant data	Field	Local Environment
	Realtime process data (e.g. current order & sensors)	Capture relevant data	Live	Process
	Accumulated process data (e.g. quality reports)	Capture relevant data	Output	
	Historic device condition	Transmit relevant data	History	Device
	Current device condition	Transmit relevant data	State	
	State of product's field	Connect to external API	Field	Global Environment
	Dashboard configuration	Low-code/No-code environment	Parameters	Configuration
	Fully aggregated and enhanced product condition	Data preprocessing	Information	Result
	Derived measures/notifications	Rule engine		
Create Digital Product Image	Usage data	Capture relevant data	Live	Process
	Product configuration	Connect to business IT-System	Configuration	Device
	Maintenance history	Connect to business IT-System	History	
	Accumulated device data (e.g. uptime, errors)	Transmit relevant data		
	Individualized analysis configuration	Low-code/No-code environment	Transformation	Configuration
	Meta data	Device metadata collection	Lateral	Asset
	Manufacturing history	Connect to business IT-System	Product	
Deliver Consumables/Supplies	Raw data about consumables usage or remaining level	Data ingestion	Field	Local Environment
	Information on the quantity of consumables in stock	Connect to business IT-System		
	Used consumables and produced scrap	Data ingestion	Output	Process
		Transmit relevant data		
	Configuration data for re-supply	Control panel	Return	Configuration
	Replenishment order	Data computation	Action	Result
	Replenishment order validation	Control panel	Information	
Derive New Products/Services	Existing product config – market data	Connect to internal platform	Configuration	Device
	Analytics web platform (e.g. Google analytics)	Connect to external API	Market	Global Environment
	Usage behavior analysis results	Connect to business IT-System	Lateral	Asset
	Customer and reseller feedback	Database storage	Product	
	Aggregated data translated into a format suited for analysis	Data preprocessing	Information	Result
	Analysis results	Data clustering	Learning	
		Data exploration		

UC	Data element	Constituting task	Cluster	Type
Improve Products/Services	Resellers/end-users feedback	Forms	Feedback	Local Environment
	Product's performance	Transmit relevant data	Output	Process
	Product's configuration	Transmit relevant data	Configuration	Device
	Usage behavior data	Connect to internal platform	Lateral	Asset
	Customer and reseller feedback	Connect to external API	Product	
	Existing product configurations	Connect to business IT-System		
	Aggregated data translated into a format suited for analysis	Data preprocessing	Information	Result
	Analysis result (identified opportunities)	Data clustering	Learning	
Offer Data Analytics	User data & feedback	Connect to external API	Feedback	Local Environment
	State of product's surroundings	Capture relevant data	Field	
	Processual parameters (e.g. programming, configuration)	Capture relevant data	Input	Process
	Processual data (e.g. temperature, energy usage, time, worker input)	Capture relevant data	Live	
	Processual results (e.g. produce quality, scrap)	Capture relevant data	Output	
	Log of productivity & maintenance	Connect to external API	History	Device
	Market data (e.g. competitive pricing, customer segments, shortages)	Connect to external API	Market	Global Environment
	Partner's data (e.g. suppliers manufacturing data, retailers demand)	Connect to external API	Supply chain	
	Analysis configuration	Scripting	Transformation	Configuration
	Results of data analysis	Data exploration	Information	Result
Offer Subscription	Billing information	Connect to business IT-System	Master Data	Customer
	Product usage / performance	Transmit relevant data		
	User feedback	Forms	Feedback	Local Environment
	Product usage / performance	Transmit relevant data	Output	Process
	Payment metric configurations	Control panel	Parameters	Configuration
	Usage data	Connect to internal platform	Lateral	Asset
	Bill	Connect to business IT-System	Action	Result
	Value delivery quantification	Data computation	Information	
Optimize Operation	State of product's surroundings	Capture relevant data	Field	Local Environment
	Process parameters	Capture relevant data	Input	Process
	Measurements during operation	Capture relevant data	Live	
	Quality assurance data	Capture relevant data	Output	
	Historical product data	Connect to external API	History	Device

UC	Data element	Constituting task	Cluster	Type
		Connect to internal platform		
	Analysis configuration	Scripting	Transformation	Configuration
	Derived recommendations	Data computation	Information	Result
	Clustered usage & product data	Data clustering	Learning	
	Predefined recommendations	Data exploration		
Provide Predictive Maintenance	Device's processual performance (e.g. temperature, energy usage, timing)	Data preprocessing	Live	Process
		Capture relevant data		
	Historical maintenance data	Connect to internal platform	History	Device
	Device's current operational state	Data preprocessing	State	
		Capture relevant data		
	Catalogue of action	Low-code/No-code environment	Knowledge	Asset
	Historical manufacturing data	Connect to business IT-System	Product	
	RUL	Data computation	Information	Result
	Derived Measure	ML model training	Information	
		Rule engine		
Update Products	Software & Hardware version administrated by system	Data ingestion	Configuration	Device
		Transmit relevant data		
		Connect to business IT-System		
		Connect to business IT-System		
	Software & Hardware version	Transmit relevant data		
	Scheduling plan	Control panel	Parameters	Configuration
	Testing groups	Rule engine	Lateral	Asset
	Firmware / Software	File storage	Product	
	Need for update	Rule engine	Information	Result
Upgrade Products	Service plan / contractual details	Connect to internal platform	Contract	Customer
	Process parameters	Capture relevant data	Input	Process
	Usage and sensor data	Capture relevant data	Live	
	Software & Hardware version	Transmit relevant data	Configuration	Device
	Product usage and maintenance history	Capture relevant data	History	
	Upgrading plan	Control panel	Parameters	Configuration
	Firmware / Software	File storage	Product	Asset
	Need for exchange product	Rule engine	Information	Result
	Opportunity for upgrade	Data exploration		
	Upgrade eligibility	Rule engine		

Blueprints – Deployment View

Table 13 *Blueprints - Deployment view (Technology collection)*

Technology	Constituting function	Description (Type)	Rollout	Level
AWS API Gateway	Actuate on command, Data clustering, Device metadata collection, Transmit relevant data	Communication (Service)	PaaS	Dev
AWS CloudFront	Dashboard, Data serving, Low-code/No-code environment	Content Delivery Network (Service)	SaaS	DevSecOps
AWS Cognito	Dashboard	Authentication/authorization (Service)	PaaS	DevSec
AWS DynamoDB	Data computation, Data preprocessing, Data serving, Database storage, Scripting	Database (NoSQL) (Service)	PaaS SaaS	DevOps
AWS EC2 (Elastic Cloud)	Automation, Data computation, ML model training, Rule engine, Scripting	Virtual machines (Service)	IaaS PaaS	DevSecOps
AWS Elastic Beanstalk	Control panel, Dashboard, Data exploration, Forms, Low-code/No-code environment, Notification, Scripting	Web App delivery (Service)	PaaS	Dev
AWS Elastic Compute Cloud Container Service	ML model training	Containers & management (Service)	PaaS	DevSecOps
AWS Elastic File System	File storage	File storage (Service)	PaaS	DevOps
AWS Elastic Load Balancer	Data ingestion, Provide API to serve RPC/data	Elastic Load Balancer (Service)	PaaS	DevSecOps

Technology	Constituting function	Description (Type)	Rollout	Level
AWS FreeRTOS	Actuate on command, Automation, Capture relevant data, Connect to business IT-System, Connect to external API, Connect to internal platform, Data computation, Data ingestion, Data preprocessing, Database storage, File storage, Receive data, Rule engine, Scripting, Transmit relevant data	Edge computing (Software)	On-Site	DevSecOps
AWS Glacier	File storage	File storage (Service)	PaaS	DevOps
AWS Greengrass	Actuate on command, Automation, Capture relevant data, Connect to business IT-System, Connect to external API, Connect to internal platform, Data computation, Data ingestion, Data preprocessing, Database storage, File storage, Receive data, Rule engine, Scripting, Transmit relevant data	Edge computing (Software)	PaaS	DevSecOps
AWS IAM	Dashboard, Data clustering	Authentication/authorization (Service)	SaaS	DevSec
AWS IoT Core	Data ingestion, Data serving, Device metadata collection, Remote update, Rule engine	IoT Data dispatcher (Service)	PaaS	Dev
AWS Lambda	Automation, Data computation, Data preprocessing, Rule engine, Scripting	Serverless functions (Service)	PaaS	DevSecOp s
AWS Recognition	Data computation, Data exploration	AI (Service)	SaaS	DevOps

Technology	Constituting function	Description (Type)	Rollout	Level
AWS Relational Database Service	Data computation, Data preprocessing, Data serving, Database storage, Scripting	Database (SQL) (Service)	PaaS SaaS	DevSecOps
AWS Route 53	Data ingestion, Data serving, Provide API to serve RPC/data, Remote update	DNS (Service)	SaaS	DevSec
AWS S3 (Simple Storage Service)	Control panel, Dashboard, File storage, Forms, Low-code/No-code environment, Notification, Provide API to serve RPC/data, Remote update	File storage (storage or web front-end publishing) (Service)	PaaS	DevSecOps
AWS SageMaker	Automation, Data clustering, Data computation, Data exploration, Data preprocessing, Low-code/No-code environment, ML model training	AI (Service)	SaaS	DevOps
AWS SimpleDB	Data computation, Data preprocessing, Data serving, Database storage, Scripting	Database (NoSQL) (Service)	SaaS	DevOps
Amazon Quicksight	Connect to business IT-System, Connect to internal platform, Dashboard, Data clustering, Data computation, Data exploration, Data preprocessing, Low-code/No-code environment, ML model training, Notification, Scripting	BI tool (Service)	SaaS	Power user
Amazon Redshift	Data computation, Data preprocessing, Data serving, Database storage, Scripting	Database (Service)	PaaS	DevOps
Azure API Management	Actuate on command, Data clustering, Device metadata collection, Transmit relevant data	Communication (Service)	PaaS	Dev

Technology	Constituting function	Description (Type)	Rollout	Level
Azure Active Directory	Dashboard, Data clustering	Authentication/authorization (Service)	PaaS	DevSec
Azure App Services and Cloud Services	Control panel, Dashboard, Data exploration, Forms, Low-code/No-code environment, Notification, Scripting	Web App delivery (Service)	PaaS	DevSecOps
Azure Archive Blob Storage	File storage	File storage (Service)	PaaS	DevOps
Azure Blob storage	Control panel, Dashboard, File storage, Forms, Low-code/No-code environment, Notification, Provide API to serve RPC/data, Remote update	File storage (storage or web front-end publishing) (Service)	PaaS	DevSecOps
Azure Cognitive Services	Data computation, Data exploration	AI (Service)	SaaS	DevOps
Azure Content Delivery Network	Dashboard, Data serving, Low-code/No-code environment	Content Delivery Netwrok (Service)	SaaS	DevSecOp s
Azure Cosmos DB	Data computation, Data preprocessing, Data serving, Database storage, Scripting	Database (NoSQL) (Service)	PaaS SaaS	DevOps
Azure DNS	Data ingestion, Data serving, Provide API to serve RPC/data, Remote update	DNS (Service)	SaaS	DevSec
Azure Database	Data computation, Data preprocessing, Data serving, Database storage, Scripting	Database (SQL) (Service)	PaaS SaaS	DevOps
Azure File System	File storage	File storage (Service)	PaaS	DevOps

Technology	Constituting function	Description (Type)	Rollout	Level
Azure Functions	Automation, Data computation, Data preprocessing, Rule engine, Scripting	Serverless functions (Service)	PaaS	DevSecOps
Azure IoT Central	Connect to external API, Connect to internal platform, Dashboard, Data exploration, Data ingestion, Data preprocessing, Database storage, Device metadata collection, Low-code/No-code environment, Notification	IoT Platform (Software)	SaaS	Power user
Azure IoT Edge	Actuate on command, Automation, Capture relevant data, Connect to business IT-System, Connect to external API, Connect to internal platform, Data computation, Data ingestion, Data preprocessing, Database storage, File storage, Receive data, Rule engine, Scripting, Transmit relevant data	Edge computing (Software)	On-Site PaaS	DevSecOps
Azure IoT Hub	Data ingestion, Data serving, Device metadata collection, Remote update, Rule engine	IoT Data dispatcher (Service)	PaaS	Dev
Azure Kubernetes Service	ML model training	Containers & management (Service)	PaaS	DevSecOps
Azure Load Balancer	Data ingestion, Provide API to serve RPC/data	Elastic Load Balancer (Service)	PaaS	DevSecOps
Azure Machine Learning	Automation, Data clustering, Data computation, Data exploration, Data preprocessing, Low-code/No-code environment, ML model training	AI (Service)	SaaS	DevOps

Technology	Constituting function	Description (Type)	Rollout	Level
Azure Table Storage	Data computation, Data preprocessing, Data serving, Database storage, Scripting	Database (NoSQL) (Service)	SaaS	DevOps
Azure Virtual Machines	Automation, Data computation, ML model training, Rule engine, Scripting	Virtual machines (Service)	IaaS PaaS	DevSecOps
ElasticSearch	Database storage, File storage	Database (cache) (Software)	On-Site IaaS PaaS SaaS	DevSecOps
GitHub, GitLab, GitX	File storage	Code hosting platform for version control and collaboration (Software)	SaaS	DevSecOps
Google App Engine	Control panel, Dashboard, Data exploration, Forms, Low-code/No-code environment, Notification, Scripting	Web App delivery (Service)	PaaS	DevSecOps
Google Cloud DNS	Data ingestion, Data serving, Provide API to serve RPC/data, Remote update	DNS (Service)	SaaS	DevSec
Google Cloud Datastore/Bigtable	Data computation, Data preprocessing, Data serving, Database storage, Scripting	Database (NoSQL) (Service)	SaaS	DevOps
Google Cloud Functions	Automation, Data computation, Data preprocessing, Rule engine, Scripting	Serverless functions (Service)	PaaS	DevSecOps
Google Cloud Load Balancing	Data ingestion, Provide API to serve RPC/data	Elastic Load Balancer (Service)	PaaS	DevSecOps
Google Cloud SQL	Data computation, Data preprocessing, Data serving, Database storage, Scripting	Database (SQL) (Service)	PaaS SaaS	DevOps
Google Cloud Storage Nearline	File storage	File storage (Service)	PaaS	DevOps

Technology	Constituting function	Description (Type)	Rollout	Level
Google Cloud storage	Control panel, Dashboard, File storage, Forms, Low-code/No-code environment, Notification, Provide API to serve RPC/data, Remote update	File storage (storage or web front-end publishing) (Service)	PaaS	DevSecOps
Google Compute Engine	Automation, Data computation, ML model training, Rule engine, Scripting	Virtual machines (Service)	IaaS PaaS	DevSecOps
Google Kubernetes Engine	ML model training	Containers & management (Service)	PaaS	DevSecOps
Grafana	Connect to business IT-System, Connect to internal platform, Dashboard, Data clustering, Data computation, Data exploration, Data preprocessing, Low-code/No-code environment, ML model training, Notification, Scripting	BI/visualization tool (Software)	On-Site IaaS PaaS	Power user
Grafana Cloud	Connect to business IT-System, Connect to internal platform, Dashboard, Data clustering, Data computation, Data exploration, Data preprocessing, Low-code/No-code environment, ML model training, Notification, Scripting	BI/visualization tool (Service)	SaaS	DevSecOps
HTTP(S)/REST	Connect to business IT-System, Connect to external API, Provide API to serve RPC/data	Protocol (Protocol)	On-Site IaaS PaaS	DevSec
InfluxDB Cloud	File storage	Database (Time series) (Service)	SaaS	Dev

Technology	Constituting function	Description (Type)	Rollout	Level
InfluxDB OSS	File storage	Database (Time series) (Software)	On-Site IaaS PaaS	DevSecOps
Kibana	Connect to business IT-System, Connect to internal platform, Dashboard, Data clustering, Data computation, Data exploration, Data preprocessing, Low-code/No-code environment, ML model training, Notification, Scripting	BI tool (Software)	On-Site IaaS PaaS SaaS	DevOps
MQTT/AMQP	Connect to internal platform	Protocol (Protocol)	On-Site IaaS PaaS	DevSecOps
Microsoft Forms	Forms	Low-code/No-code forms handling (Service)	SaaS	Power user
Microsoft PowerBI Cloud	Connect to business IT-System, Connect to internal platform, Dashboard, Data clustering, Data computation, Data exploration, Data preprocessing, Low-code/No-code environment, ML model training, Notification, Scripting	BI tool (Service)	On-Site IaaS PaaS	Power user
Microsoft PowerBI Desktop	Connect to business IT-System, Connect to internal platform, Dashboard, Data clustering, Data computation, Data exploration, Data preprocessing, Low-code/No-code environment, ML model training, Notification, Scripting	BI tool (Software)	On-Site IaaS PaaS	Power user
MongoDB	Database storage	Database (NoSQL) (Software)	On-Site IaaS PaaS SaaS	DevSecOps

Technology	Constituting function	Description (Type)	Rollout	Level
Neo4J	Database storage	Database (Graph) (Software)	On-Site IaaS PaaS SaaS	DevSecOps
NodeJS	Automation, Scripting	Programming language (Software)	On-Site IaaS PaaS	DevSecOps
NodeRed	Automation, Low-code/No-code environment, Scripting	Programming tool (Software)	On-Site IaaS PaaS	Dev
OneDrive	File storage	File storage (Service)	SaaS	User
Pepite DATAmaestro	Automation, Connect to business IT-System, Dashboard, Data clustering, Data exploration, Data ingestion, Data preprocessing, Database storage, Low-code/No-code environment, ML model training, Rule engine, Scripting	Advanced analytics platform (Service)	PaaS	Power user
Python	Automation, Scripting	Programming language (Software)	On-Site IaaS PaaS	DevSecOps
Raspberry Pi	Actuate on command, Data ingestion, Receive data, Transmit relevant data	Digitally enabling a physical product (Product)	On-Site	DevSecOps
Synapse Analytics	Data computation, Data preprocessing, Data serving, Database storage, ML model training, Scripting	Database & analytics (Service)	PaaS SaaS	DevOps

Technology	Constituting function	Description (Type)	Rollout	Level
Tableau (Server)	Connect to business IT-System, Connect to internal platform, Dashboard, Data clustering, Data computation, Data exploration, Data preprocessing, Low-code/No-code environment, ML model training, Notification, Scripting	BI tool (Service)	On-Site IaaS PaaS	DevSecOps
Tableau Cloud	Connect to business IT-System, Connect to internal platform, Dashboard, Data clustering, Data computation, Data exploration, Data preprocessing, Low-code/No-code environment, ML model training, Notification, Scripting	BI tool (Software)	SaaS	Power user
Tinkerforge Bricklets	Capture relevant data, Transmit relevant data	Hardware (Product)	On-Site	Dev
myopenfactory	Connect to business IT-System, Connect to external API, Provide API to serve RPC/data	IoT Platform (Service)	PaaS	DevOps

Blueprints – Business Model Canvas

Table 14 *Blueprints – Business Model Canvases*

UC	Business Model Element	Cluster
Analyse Usage Behaviour	Better understanding of the customer's problems to derive appropriate measures	Value Proposition
	Implement capture of missing data	Key Activities
	Continuous benefit analysis to derive strategical data collection needs	
	Analyze data to derive required information	
	Usage data	Key Resources
	Data Analyst	Key Partners
	Operations: communication, payload storage, infrastructure, …	Cost Structure
	Watchdog / Surveillance of customers actions (negative connotation, especially in Europe)	Customer Relationship
	(potential) Individual contact	Channels
	(potential) Data platform	
	Products that have a significant degree of user interaction or parametrization	Customers Segments
	(potential) Potential to realize other use cases (update product, upgrade product)	Revenue Streams
Assist Operation	Focusing on creating value with the product instead of learning/handling the product	Value Proposition
	Helping to correctly use the product	
	Making a complex product manageable	
	Making a product available for a broader user segment	
	Analyze customer behavior	Key Activities
	Structure usage recommendations regarding appliance and user groups	
	Effectively communicate assistance to user	
	Expertise in own product	Key Resources
	Analyzed and structured data sourced from knowledge management (all the assets created to support product usage)	
	Usage Data	
	Servicing center	Key Partners
	Expert in UI/UX design & user behavior	
	(potential) Expert in user psychology	
	Most expenditure within supporting processes (maintaining data / recommendation)	Cost Structure
	Expert for questions / assistance regarding the product	Customer Relationship
	Helping hand / guard of customer	
	Customer Support	Channels
	Product-interfaces	
	Products with high level of parameter complexity	Customers Segments
	Segment of industry when machines are operated by workers	
	Modern / new products with limited expertise	
	Large scale product with few uses	
	Individual Customer Support	Revenue Streams
	Product configuration: fee for smart assistance / 24/7 servicing	
Condition Monitoring	Gain transparency into the historical and real time operation	Value Proposition
	Provide a health status of the different components of a smart product to guarantee the intended OEE	
	Warn about potential misuses	Key Activities
	Monitor sensors values	
	Data collected	Key Resources
	Set of sensors (physical or virtual)	

UC	Business Model Element	Cluster
	(potential) More effective customer service (due to better understanding of the customer need and actions to fulfill that need)	Cost Structure
	(potential) Maintenance expenses reduction	
	Maintenance	Customer Relationship
	Product-related servicing	Channels
	(potential) Monitoring platform	
	Direct sales of extra option on the SP	
	B2B/B2C	Customers Segments
	One-time fee extra pack	Revenue Streams
	Extra service (subscription-based)	
Create Digital Product Image	Capturing a digital image of a product -> monitoring and documenting a product lifecycle	Value Proposition
	Structured and accessible product data	
	Having a digital copy of the product, containing all relevant data, that is digitally interfaceable	
	(potential) Base for further use cases related to data of individual products	Value Proposition
	Capturing relevant data	Key Activities
	Structuring and persisting data	
	Collected Data	Key Resources
	Experts for Data Stores	Key Partners
	Experts for Monitoring / Visualization / UI	
	Technical experts for capturing relevant data in physical environment	
	Maintaining Data Stores	Cost Structure
	Data Dependency	Customer Relationship
	(potential) Further use cases based on data (monitoring, analysis, servicing)	Channels
	Possibly Product Data Inspector Application	
	Highly configurable product developers	Customers Segments
	Mainly maintenance heavy product applications, where data is supposed to be accessible	
Deliver Consumables/Supplies	Reducing warehouse utilization	Value Proposition
	Focusing on the value the product creates instead of maintaining the product	
	No hassle of keeping track of stock and reordering at the correct time	
	Analyzing use and predicting restocking periods	Key Activities
	Tracking Consumables / Supplies of product in field	
	Just-in-time delivery of goods and its optimization	
	Product use data	Key Resources
	Network of customers	
	Supplier of goods	Key Partners
	Data Analytics (prediction & travelling salesman)	
	(potential) Expenditures of optimizing operation	Cost Structure
	Handling and delivery of goods	
	Supplier of restockable items	Customer Relationship
	Recurring postal service	Channels
	(potential) Servicing interface to manage/tune subscription	
	Products that have designated degradable parts	Customers Segments
	Products that consume supplies/materials	
	Pay-per-use / recurring purchases	Revenue Streams

UC	Business Model Element	Cluster
Derive New Products/Services	Understanding of needs and expert in customer's branch	Value Proposition
Derive New Products/Services	Availability of new products and services that are (better) tailored to the customer's needs	Value Proposition
Derive New Products/Services	Development of solutions to address specific needs	Key Activities
Derive New Products/Services	Understanding the customers needs	Key Activities
Derive New Products/Services	Customer relationships	Key Resources
Derive New Products/Services	Usage data	Key Resources
Derive New Products/Services	Data Analytics	Key Partners
Derive New Products/Services	Technology Specialist	Key Partners
Derive New Products/Services	Branch Experts (even customers themselves)	Key Partners
Derive New Products/Services	Product / Service development	Cost Structure
Derive New Products/Services	(potential) Specialization can create risk of dependency	Cost Structure
Derive New Products/Services	(potential) Potential of external cost coverage (research program & investors)	Cost Structure
Derive New Products/Services	Vendor is seen as a supplier of specific solutions	Customer Relationship
Derive New Products/Services	Vendor is seen as an expert in the customer's branch	Customer Relationship
Derive New Products/Services	Sales/Distribution	Channels
Derive New Products/Services	Service	Channels
Derive New Products/Services	Branches, where there is still a concrete solution to be found to a problem without a workaround (upward potential)	Customers Segments
Derive New Products/Services	Customers with high complexity needs	Customers Segments
Derive New Products/Services	Premium on specific solutions (USP: lone specialist for branch)	Revenue Streams
Derive New Products/Services	Sale of new products	Revenue Streams
Improve Products/Services	Lesser chance of outdating products leading to replacements/write-offs	Value Proposition
Improve Products/Services	Cost efficient renewal of provided services/benefits	Value Proposition
Improve Products/Services	Ever improving product after sale	Value Proposition
Improve Products/Services	Development of additive solutions	Key Activities
Improve Products/Services	DevSecOps pipeline management	Key Activities
Improve Products/Services	Surveying product related problems, needs and potentials	Key Activities
Improve Products/Services	Usage data and customer feedback	Key Resources
Improve Products/Services	Strong and active development base	Key Resources
Improve Products/Services	(potential) Software Development Partner	Key Partners
Improve Products/Services	(potential) Technological Expert	Key Partners
Improve Products/Services	Constant product / service development	Cost Structure
Improve Products/Services	Concurrent active involvement in solving customer-relevant problems	Customer Relationship
Improve Products/Services	Constant supplier of product improvements	Customer Relationship
Improve Products/Services	Service	Channels
Improve Products/Services	Software/Update Centers	Channels
Improve Products/Services	Fast developing and mainly unchartered branches	Customers Segments
Improve Products/Services	High-active-use products	Customers Segments
Improve Products/Services	User products with complex graphical interfaces (with a view to simplification through redesign)	Customers Segments
Improve Products/Services	Products with high modularity	Customers Segments
Improve Products/Services	(potential) Creation or redesign of new modules and/or paying options in the product	Revenue Streams
Improve Products/Services	(potential) Premium Upgrades / Services	Revenue Streams
Offer Data Analytics	Provide value-added information or proof about the functioning of a smart product or its environment (historical data, computing-based data)	Value Proposition
Offer Data Analytics	Providing data insights, visualization	Key Activities
Offer Data Analytics	Collected data	Key Resources
Offer Data Analytics	Visualization means	Key Resources
Offer Data Analytics	(potential) Analysis environment	Key Resources
Offer Data Analytics	Co-development with data scientists	Key Partners
Offer Data Analytics	(potential) Subscription price for a visualization platform + develop API connection	Cost Structure

UC	Business Model Element	Cluster
	Computing time if backend-based	
	Customer bonding, addiction	Customer Relationship
	Data dependency	
	Product-related services	Channels
	(potential) Data Analytics platform for customer driven analysis and visualization	
	Data-driven or data-rich branches (requires individual evaluation)	Customers Segments
	One-time extra fee (B2C product)	Revenue Streams
	Subscription model for service to end-users	
	(potential) Cross-selling (BBK case)	
Offer Subscription	Higher flexibility in service cancellation	Value Proposition
	Reduced investment cost	
	Ensure operation and functionality of product	Key Activities
	Proactive servicing	
	Optimize own operation to increase margins	
	High initial investment (not charged to customer)	Cost Structure
	Servicing cost	
	Contractual obligation	Customer Relationship
	Service / Sales	Channels
	Products that usually require high investment cost: production / office machines	Customers Segments
	Concurrent contractually bound payments	Revenue Streams
Optimize Operation	Having the existing product fulfil the individual needs better	Value Proposition
	Operating the product at best economical, conservative, environmental or productive state	
	Analyzing product data with respect to pursued goal	Key Activities
	D/Refine model for understanding smart product operation	
	Take action based on analysis results	
	Understanding the customers individual needs	
	In-depth understanding of product's effectual correlations / causalities	Key Resources
	Collected Data	
	(potential) Data Scientist	Key Partners
	Maintenance expenses can become a part of individual optimization	Cost Structure
	(potential) More effective customer service (due to better understanding of the customer need and actions to fulfill that need)	
	Helping customers achieving their individual goals in the most optimal way	Customer Relationship
	Reducing the effort in optimizing product use	
	Branches that incorporate products with economical, environmental, or production side effects	Customers Segments
	Branches that incorporate products that fulfill a quantifiable use (e.g. machines in a production process)	
	(potential) One-time fee or subscription-based service	Revenue Streams
	(potential) Increased trust / favorability by customer to offer further products / services	
Provide Predictive Maintenance	Improving reliability and up-time of product	Value Proposition
	Reduce unnecessary costly maintenance	
	Optimize product operation	
	Detect creeping maintenance cases	
	Capturing and analyzing product usage data	Key Activities
	Enhancing product insights from customer network	
	In-depth understanding of product	Key Resources
	Usage data	

UC	Business Model Element	Cluster
	Data Analytics Expert	Key Partners
	Expenditures in refining predictive modelling and gaining product understanding	Cost Structure
	Operating data analytics platform	
	Guardian for ensuring operation of the product	Customer Relationship
	Application accompanying product	Channels
	Customer service	
	Products with high uptime demands	Customers Segments
	Products with periodical maintenances	
	Customers with small margins	
Update Products	Lesser chance of outdating products leading to replacements/write-offs	Value Proposition
	Cost efficient renewal of provided services/benefits	
	Ever improving product after sale	
	Development of additive solutions	Key Activities
	DevSecOps pipeline management	
	Surveying product related problems, needs and potentials	
	Usage data and customer feedback	Key Resources
	Strong and active development base	
	(potential) Software Development Partner	Key Partners
	(potential) Technological Expert	
	Constant product / service development	Cost Structure
	Concurrent active involvement in solving customer-relevant problems	Customer Relationship
	Constant supplier of product improvements	
	Service	Channels
	Software/Update Centers	
	Fast developing and mainly unchartered branches	Customers Segments
	High-active-use products	
	User products with complex graphical interfaces (with a view to simplification through redesign)	
	Products with high modularity	
	(potential) Creation or redesign of new modules and/or paying options in the product	Revenue Streams
	(potential) Premium Upgrades / Services	
Upgrade Products	Lesser chance of outdating products leading to replacements/write-offs	Value Proposition
	Cost efficient renewal of provided services/benefits	
	Ever improving product after sale	
	Development of additive solutions	Key Activities
	DevSecOps pipeline management	
	Surveying product related problems, needs and potentials	
	Usage data and customer feedback	Key Resources
	Strong and active development base	
	(potential) Software Development Partner	Key Partners
	(potential) Technological Expert	
	Constant product / service development	Cost Structure
	Concurrent active involvement in solving customer-relevant problems	Customer Relationship
	Constant supplier of product improvements	
	Service	Channels
	Software/Update Centers	
	Fast developing and mainly unchartered branches	Customers Segments
	High-active-use products	
	User products with complex graphical interfaces (with a view to simplification through redesign)	
	Products with high modularity	

UC	Business Model Element	Cluster
	(potential) Creation or redesign of new modules and/or paying options in the product	Revenue Streams
	(potential) Premium Upgrades / Services	

Additional Content for Blueprints – Transversal Requirements

Introduction

The objective of this document is to present the requirements that are applicable to all "Smart Product" architectures that allow for the implementation of use cases.

These requirements have deliberately not been integrated into each of the use cases or blueprints.

These transversal requirements are to be considered when designing the architecture. They apply for the most part to all the functional elements of the architecture, even if some, such as usability/UX, are more specific to the UI cluster.

Cybersecurity

Cybersecurity is one of the most important requirements. Moreover, it is nowadays more than ever relevant to have a "by design" approach, which means considering the challenges from the very beginning of the development in order to control the costs and potential risks as much as possible. In this respect, we invite you to take note of the different points below that should be kept in mind in the development, testing and maintenance process of the solution:

• Proactive Security Approach: By integrating cybersecurity measures into the design phase, organizations can adopt a proactive approach to security. This means considering potential threats, vulnerabilities, and attack vectors from the outset and implementing appropriate safeguards. Cybersecurity “by design” helps identify and address security gaps early, reducing the likelihood of successful attacks or breaches in the future.

• Protection Throughout the Product Lifecycle: Smart products have a long lifecycle, including development, deployment, operation, and eventual decommissioning. A cybersecure by design architecture ensures that security considerations are incorporated at every stage. This protects the product and its users from evolving cybersecurity threats throughout its entire lifespan.

• Enhanced Resilience: Cybersecure architecture by design improves the resilience of smart products against cyberattacks. It involves implementing mechanisms such as secure communication protocols, encryption, access controls, and secure software development practices. These measures make it more difficult for attackers to compromise the product, maintain persistence, or manipulate its functionalities.

• User Trust and Confidence Cybersecurity “by design” demonstrates a commitment to protecting user data (e.g. GDPR-related), privacy, and safety. By prioritizing security from the outset, organizations can build trust with their customers, enhance brand reputation, and encourage wider adoption of their smart products.

• Compliance with Regulations: IoT products often handle sensitive user data and interact with various networks and devices. Many countries and regions have introduced regulations and standards related to cybersecurity and data protection. A cybersecure “by design” architecture helps organizations comply with these regulations. It ensures that products meet the necessary security requirements, reducing legal and financial risks associated with non-compliance.

• Cost and Time Efficiency: Addressing security issues after a product has been developed can be time-consuming and costly. In contrast, integrating cybersecurity by design streamlines the development process. It avoids the need for significant modifications or rework later on, saving both time and resources. It also minimizes the potential costs associated with security breaches, such as data loss, damage to reputation, or legal liabilities.

In this context, it is also relevant to take note of the European Commission's draft regulation called “Cyber Resilience Act” (https://digital-strategy.ec.europa.eu/en/library/cyber-resilience-act) on products integrating digital elements. Certain security features are about to become mandatory and a period of transition is foreseen.

Scalability

It is always necessary to think about the potential evolution of the architecture as a result of the increase in the number of smart products, the type of applications that will be deployed and the conquest of new markets where the regulation may be different. Of course, it is difficult to foresee everything, but having to review part or all of an architecture is not easy.

The cost perspective should also be taken into account, such as the marginal cost of a smart product that has just been commissioned or the cost of a new user using the service.

Observability / monitoring

Smart products often involve complex interactions between hardware, software, and connectivity components. Observability provides vendors with the means to detect and debug issues more effectively. By monitoring various metrics, logs, and traces, they can pinpoint the root cause of problems, troubleshoot them efficiently, and release timely updates or patches.

Usability/User Experience (speed, mobility)

When delivering a smart product, it is important to consider both usability and user experience. This involves designing the product with the user in mind, considering their needs, goals, and preferences. It also involves testing the product with real users to identify any usability or UX issues and iterating on the design to address these issues. By prioritizing usability and UX design, smart product vendors can create products that are not only technically advanced but also user-friendly, enjoyable, and satisfying to use.

Additional content for blueprints – Architecture Management

Enterprise Architecture Management (EAM) is a discipline that focuses on managing and aligning an organization’s business and technology components to achieve its strategic goals. It involves the creation, documentation, analysis, and evolution of an enterprise architecture framework to ensure that all aspects of the organization are designed, integrated, and optimized effectively. Here are the key steps involved in the process of enterprise architecture management:

1. Establishing the Scope: Define the scope of the enterprise architecture effort, including the boundaries, objectives, and stakeholders involved. This step involves understanding the organization’s business strategy, goals, and key drivers.

2. Current State Assessment: Assess the organization's current architecture, including its business processes, applications, data, and technology infrastructure. This involves gathering information through interviews, workshops, documentation review, and analysis of existing systems.
3. Defining the Target State: Develop a vision for the future state of the organization's architecture that aligns with its strategic objectives. Identify the desired outcomes and benefits of the target architecture and define the principles and standards that will guide its development.
4. Gap Analysis: Perform a gap analysis by comparing the current state architecture with the target state. Identify the gaps, inefficiencies, and areas of improvement that need to be addressed to move from the current state to the desired future state.
5. Roadmap Development: Develop a roadmap that outlines the sequence of initiatives and projects required to transition from the current state to the target state. Prioritize the initiatives based on their business value, feasibility, and dependencies. The roadmap should consider factors such as technology evolution, business priorities, resource availability, and organizational readiness.
6. Architecture Development: Start developing the architecture by defining the architecture components, such as business architecture, information architecture, application architecture, and technology architecture. These components should be designed to align with the organization's goals and requirements.
7. Implementation and Governance: Implement the architecture components in a phased manner according to the roadmap. Establish governance processes to ensure that the architecture is followed, and any deviations are managed effectively. This includes monitoring the implementation progress, managing risks and issues, and measuring the outcomes against the defined goals.
8. Continuous Improvement: EAM is an ongoing process that requires continuous improvement. Regularly evaluate the effectiveness and efficiency of the architecture and make necessary adjustments to ensure it remains aligned with the changing business needs and technology landscape.

Throughout the process, effective communication and collaboration with stakeholders from different business units and IT departments are crucial. EAM aims to provide a holistic view of the organization's architecture, promote standardization and integration, improve decision-making, and enable strategic planning for business transformation and growth.

The Open Group Architecture Framework (TOGAF) is a widely used framework for enterprise architecture management. It provides a structured approach and set of best practices to guide organizations through the process of creating and managing their enterprise architecture. TOGAF works in the context of enterprise architecture management as follows:

1. Architecture Development Method (ADM): The core of TOGAF is the Architecture Development Method (ADM). It is a step-by-step process that guides the development and evolution of an enterprise architecture. The ADM consists of several phases, including Preliminary, Architecture Vision, Business Architecture, Information Systems Architecture, Technology Architecture, Opportunities and Solutions, Migration Planning, Implementation Governance, and Architecture Change Management.
2. Architecture Development Method Deliverables: At each phase of the ADM, TOGAF provides a set of deliverables or artifacts that serve as the building blocks of the enterprise architecture. These deliverables include documents, models, and diagrams that capture the current and target architectures, gap analysis, stakeholder viewpoints, and implementation plans.

3. Architecture Content Framework: TOGAF defines an Architecture Content Framework that helps organize and structure the architecture artifacts. It provides a consistent way to describe and document the architecture components, such as business processes, data models, application portfolios, and technology standards.
4. Architecture Repository: TOGAF recommends the use of an Architecture Repository to store and manage the architecture artifacts. The repository serves as a centralized knowledge base for all architecture-related information and enables traceability, versioning, and reuse of artifacts across different projects.
5. Architecture Governance: TOGAF emphasizes the importance of architecture governance to ensure the effective management and compliance of the enterprise architecture. It provides guidelines for establishing an Architecture Governance Framework, including defining roles and responsibilities, establishing decision-making processes, and monitoring the implementation of architecture standards and guidelines.
6. TOGAF Reference Models: TOGAF includes reference models that provide pre-defined templates and best practices for specific domains, such as the Technical Reference Model (TRM), the Integrated Information Infrastructure Reference Model (III-RM), and the Foundation Architecture (common services and infrastructure).
7. TOGAF Certification: TOGAF offers a certification program that validates an individual's knowledge and understanding of the framework. It provides different levels of certification, including TOGAF Foundation and TOGAF Certified, which can enhance an architect's credibility and demonstrate their proficiency in applying TOGAF.

By following the principles, methods, and guidelines provided by TOGAF, organizations can establish a structured and standardized approach to enterprise architecture management. TOGAF helps in aligning business and IT strategies, enabling effective communication and collaboration, promoting architectural consistency and quality, and supporting the organization’s overall business transformation goals.

Collaboration strategy guideline

Focus on leveraging co-development for your Smart Product journey

Authors: Thierry Coutelier, Gaëtan Hug, Sebastian Kremer
Version: 1.6

Introduction

Foreword

Collaboration consists of working with other companies (partners and/or subcontractors) to develop and possibly maintain all or part of one or more components required to bring a "smart product" to market. In this type of operation, which can help you to go to market faster, it is important to set up a framework for the proper functioning of this collaboration. This is what this document is trying to help you discover and explain. The goal is to help you develop your project in the best possible way while limiting the risk and avoiding many of the pitfalls other companies have encountered in the past.

This document is addressed to the person who wishes to know more about collaboration and sourcing in Smart Product development as well as to the person already experienced who will certainly find useful additional information. We'll specifically highlight one collaboration mode called "co-development".

In this document you will be first accompanied through the different steps of the development of a Smart Product, from the ideation and value discovery phase to the solution architecture. Then, the document will provide insights for making "Make or Buy" decisions for the different components of the technical solution – hardware and software. Furthermore, the collaboration models and co-development best practices will be described. Finally, some key insights regarding the management of IP in the context of a collaboration will then be detailed along with collaboration challenges and risks management.

What is co-development?

Here are "co-development" definitions found in the literature. It is also referred to as "co-creation" by other sources, we will though stick to the term "co-development".

> *"Co-development, a training method that relies on experience and communication with peers, and consolidates teamworking practices within the same company or separate companies." [30]*

> *"In other words, co-development is a learning approach, which favours interactions among businesses to create collective intelligence." [31]*

Following our research, the definition concluded within the BlueSAM project is more oriented towards technological development and does not only concern the ideation process. It is as follows:

> Co-development involves a collaborative and integrated approach, with each party contributing its expertise and resources to jointly develop a product or

service. Co-development agreements often involve sharing of intellectual property and a higher level of risk and reward sharing. Co-development can also be limited to the simple fact that there is a transfer of knowledge that is of strategic and economic benefit to the parties.

Co-development may be considered as a subset of partnership, as it can be one of many potential collaboration models within a broader partnership. Partnerships typically imply more strategic and holistic approach to collaboration, with each party bringing complementary capabilities, resources, and expertise to the table. In contrast, co-development is often focused on a specific project or product development initiative and may involve a more limited scope of collaboration.

This is also to be distinguished from subcontracting that usually involves an even more limited scope, with the subcontractor being responsible for a specific deliverable or outcome. The subcontractor usually has less autonomy in how the work is done and is paid a fixed fee or hourly rate for their services.

Prerequisites for a successful Smart Product Development Journey

In order to ensure the successful development of a smart product and following our field experience, two distinct profiles are needed: a market-oriented profile and a technic-oriented profile. It will most likely be difficult to function optimally if these two roles are taken on by the same person within the organization. In a start-up, this means at least a “CEO” who takes on the commercial/sales aspect and contacts with customers and a “CTO” who will take responsibility for all technical aspects while remaining in contact with customer requests relayed by the CEO.

By extension and for larger companies, these roles are filled by a "Product Manager" – CEO equivalent – and "Solution Architect" – CTO equivalent.

- The Product Manager focuses on the current business or the business to be developed. He understands the expectations of the customers in order to define the business requirements in the best possible way. He also knows how to evaluate the value of different developments for the customer and is therefore able to define the product strategy as well as possible.
- The Solution Architect defines the system requirements, evaluates the various alternatives from a cost point of view and is concerned with advancing the technical development of the solution as a whole.

The problem to be solved and the business case around it are initially identified by the Product Manager/CEO who will trigger the search for a technical solution by the Solution Architect/CTO. However, the process does not follow a linear development. After the initial interaction, the feedback coming from the Solution Architect may trigger adjustments to the business case or taking a different perspective on the problem to be solved. Multiple iterations may be required before converging towards the business case and its technical solution to be developed.

Figure 1: Main roles in Smart Product development

Focus on requirements

The starting point for the development of the solution and on which as much clarity as possible is needed are the "functional requirements", i.e.: what does the product need to do. These requirements need to be defined with enough precision to avoid misunderstandings. The context for which those are defined is as important for understanding as the functional requirement itself. As such it is of great help to anchor the functional requirements to the description of the different use cases of the Smart Product identified during the value discovery phase. When these are clear, they can easily be transcribed into requirements from a system perspective (“system requirements”). These different requirements can then be integrated into a development roadmap in the form of tasks or “stories” (agile concept – software engineering).

Product management

This role of product manager is essential for the success of the project. It needs to be filled by someone within the company who already knows the customers, their needs and is able to transcribe them faithfully. Taking on this role represents a significant investment of time. It is therefore important to ensure that this resource person is available.

If, due to a lack of time or qualified internal resources, this role has to be subcontracted or entrusted to a person in the company who does not have experience from the customer's perspective, care should be taken to transfer all the experience of the market (business) to this person progressively and to ensure that regular contact points are set up to monitor progress and to ensure that numerous customer contacts are made. However, it is highly recommended not to outsource this outside the company.

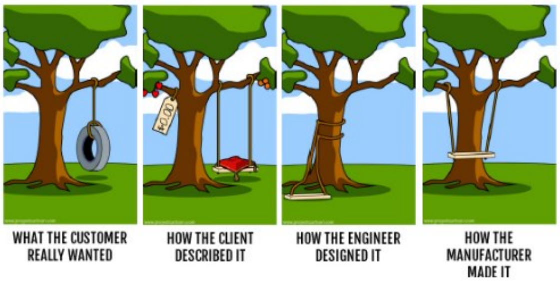

Figure 2: A rather popular illustration of the importance of listening and communicating information accurately

Finally, product management and the business knowledge is in your hands at the beginning of the project. Don't underestimate the time needed to educate internal or external people.

Solution Architecture

The role of the Solution Architect can more easily be outsourced. This person will define all the possible alternatives to meet the functional requirements while challenging the product manager to ensure that the right technical requirements are met (speed, accessibility of the service, security level, etc.).

Architecture and related documentation

Producing documentation helps the different parties to understand each other better and to formalize responses to each other's expectations. On the basis of these documents, specific architecture diagrams can be developed to cover the whole solution.

It is at this stage of the project that the BlueSAM blueprints and the associated method best help you to define a first version of a high-level architecture. All the iterations between people on this documentation are a key stage in the successful completion of developments. It is important to list all the use cases that will be covered during the lifetime of the smart product (production, support, maintenance, ...) and for all the stakeholders (external and internal). At this stage, you should not rush to choose the technologies that will support the system.

Make or Buy?

The next question is how to implement the various functional blocks. Is there already a solution on the market (off-the-shelf, SaaS - Software-as-a-Service, PaaS – Platform-as-a-Service) that would be suitable, or do we want to develop something because what exists on the market is not complete and/or flexible enough? Will the development be done in-house or outsourced? And even if it is an existing functional block on the market, how and by whom can it best be integrated into the solution? This brings us to a first point we want to cover in this document: "Make or Buy": should I develop (or get someone to develop) a custom component rather than taking advantage of an off-the-shelf solution that I can potentially customize and integrate into the smart product.

Do you need new or existing skills to develop modules or features? You will certainly ask yourself whether something that exists on the market might not be suitable for your needs, even if it means adapting it.

At some point, you may think that nothing seems to be exactly what you want and that it will most certainly have to be developed.

Moreover, do you have the skills in-house and the time to develop this? Is it strategically relevant for your business to spend some time on it?

If you opt to ‘do it yourself’ (“make” perspective) and you don’t have all the skills required, it is important to consider that it can be difficult to suddenly become a software publisher or electronics design agency and yet be able to directly follow good practice and standards in this field.

Having it developed by an external company could also work (outsourcing/agency model), but this will require a minimum of monitoring and skills to manage the collaboration.

In the end, this decision is not purely financial, but also strategical, impacts risk management and impacts the organization.

What are the factors that affect this “make or buy” decision:

- Where does the value creation reside in the solution?
- Where is our company’s differentiator/IP? Is the company able to focus and produce key differentiators (e.g. specific assets, or skills like data analysis). Is there any IP within the expertise of the company?
- What level of control is needed to adapt a system to our company’s needs and future vision? The flexibility of make vs the rigidity of buy has to be considered.
- Which dependencies does our company allow regarding a solution that is bought? -> vendor lock-in + costs to migrate to another solution (vendor stops, collaboration discontinued, …)
- What is the allowable time-to-market? -> delays + opportunity cost
- Total cost of ownership of a solution for our company? -> some costs are less visible/tangible when making (security, scalability, certification, support, changes/extensions) or buying (licensing/subscription costs, integration costs)
- Do we have (access to) development capabilities? -> not only internal but also external
- Never underestimate the vast body of existing solutions -> an important pitfall to IoT developments is poor market research

Let's take the example of an IoT platform to which smart products will connect. There are many possibilities:

- An off-the-shelf IoT platform that allows you to meet a generic collection/processing/visualization (sensor-to-cloud) need and possibly takes advantage of a no-code/low-code perspective to make your job easier. This can be complemented by various modules or add-ons to meet more specific needs. Examples in this area include Azure IoT Central or PTC Thingworx.
- A more vertical IoT platform that specifically addresses sector-specific needs. Here you will find a certain number of the usual features and you may be able to make some customizations or add a little additional development via an API or ask for a tailor-made solution to meet a more specific request.
  Examples include Televend (Croatia - automated vending machines) and Opinum (Belgium - centralized management of energy-related devices).

- Developer-centric IoT modules that fit together to offer a complete and customized solution, but which require developer skills. Here we find platforms such as AWS IoT or Azure IoT that will be extended with some other features coming from other services of the same cloud provider.

At the same time, you will also ask yourself to what extent you want to outsource the management, operations and maintenance of these components.

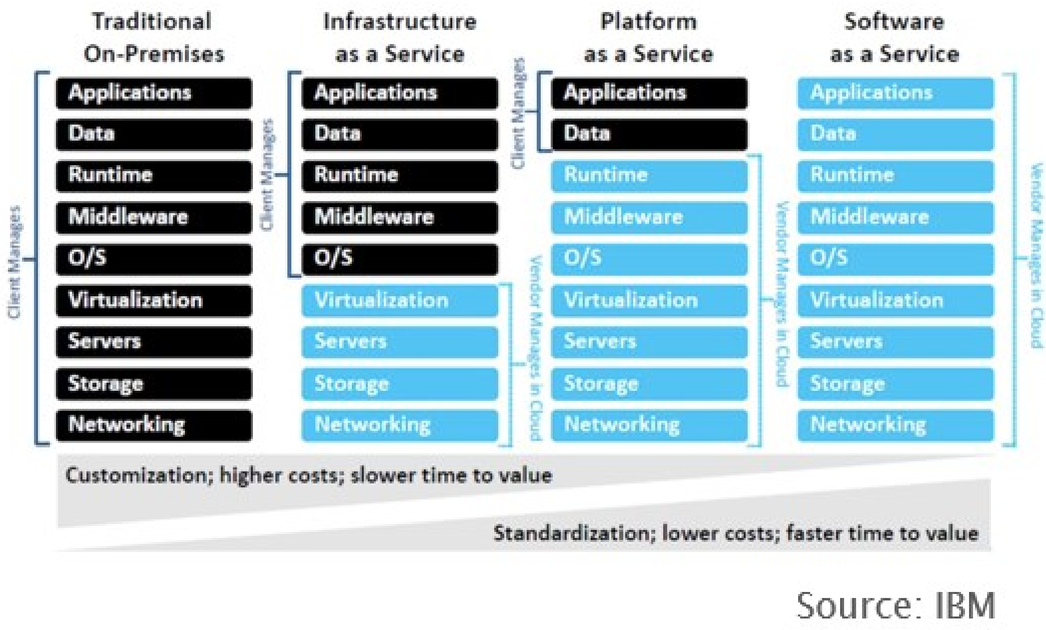

Figure 3: From on-premises to Software-as-a-Service: impact of Ops

In a traditional on-premises mode (1st option on the left), you will have to take care of all the elements that underpin the availability of your component, but this will offer you the greatest capacity for customization.

In an IaaS (Infrastructure-as-a-Service) mode, the management of the infrastructure is delegated to a subcontractor who provides you with machine availability and computing capacity in a datacenter or in the cloud.

In PaaS (Platform-as-a-Service) mode, you simply manage the applications and data and everything else is outsourced. This leaves you room for advanced customization through code within the applications running on the infrastructure.

Finally, SaaS (Software-as-a-Service) mode takes the management burden off your shoulders while offering you a solution as a service with more limited customization capabilities. These solutions have the advantage of being available quickly.

The more you move from the left to the right, the more you turn towards a cloud infrastructure with an increasing number of standardized services. The reasons that might lead you to go in this direction are:

- Time to market
- Operational Cost
- Knowledge and capabilities
- Flexibility (configuration vs customization)
- Scalability

To conclude this section, we would like to stress that it is better to focus your in-house activities on the core activities of the company: strategy, customer development, product exploration, product management, R&D management, ... In all non-core activities, your company is likely to be less effective because you do not have the skills, experience, or infrastructure to deploy this in the best way. It is this assessment that will push you to outsource developments or not.

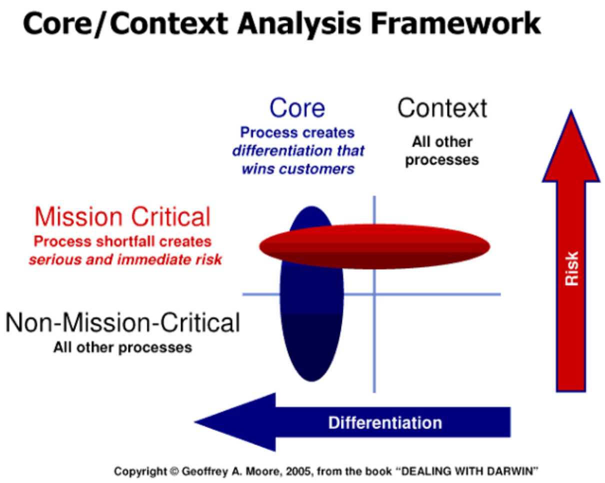

Figure 4: Reference : https://secretpmhandbook.com/the-mission-critical-corecontext-model-for-product-managers/

In this perspective, co-development has to be considered in your decision-making process:

"Co-creation is a very good match here: you retain and grow your internal knowledge, while the external experts you bring into your organization give you input on best practises at every step of the way. A good co-creation partner has the potential to bring fresh insights by putting new technologies and methodologies on the table. This enables you to innovate quickly while you continue to deliver quality." [32]

Collaboration models and ways to collaborate

Collaboration on a technical development comes in different flavors which can mainly be summarized by the following three models:

- Supplier: delivers products and services to the company
- Subcontractor: delivers services/performs specific tasks as part of a project

- Co-development partner: contributes to a business by sharing risk, cost, revenue, and/or reputation while often exchanging knowledge during the joint development of a product or service.

At all stages of product development, from ideation to improvement projects once the product is on the market, it will be necessary to consider the collaboration model and the choice of company or companies to work with.

For the most standardized and common services and products used on a one-off or recurrent basis without any particular risk, the "supplier" mode is the most suited.

For more specific and specialized tasks over the course of the project, rather of a one-off nature, such as innovation/ideation services and a feasibility study, subcontracting shall be preferred.

Finally, the co-development partnership will be privileged in the perspective of longer-term collaboration where each party will find a potential benefit while potentially sharing the risks of the venture. In the following, three different models coming from literature will be compared:

Chesbrought & Schwartz (2007) argue that for projects of an innovative nature, business models should really focus on a co-development strategy. They have created a model which aims to help finding and selecting a partner for a co-development strategy. Forming these co-development partnerships should depend on four criteria:

1. Defining the business objective
2. Assessing the capabilities you have and the ones you require
3. Determining the degree of business model alignment with partners
4. “In managing the partnership think not only of current needs but also future needs“ [33]

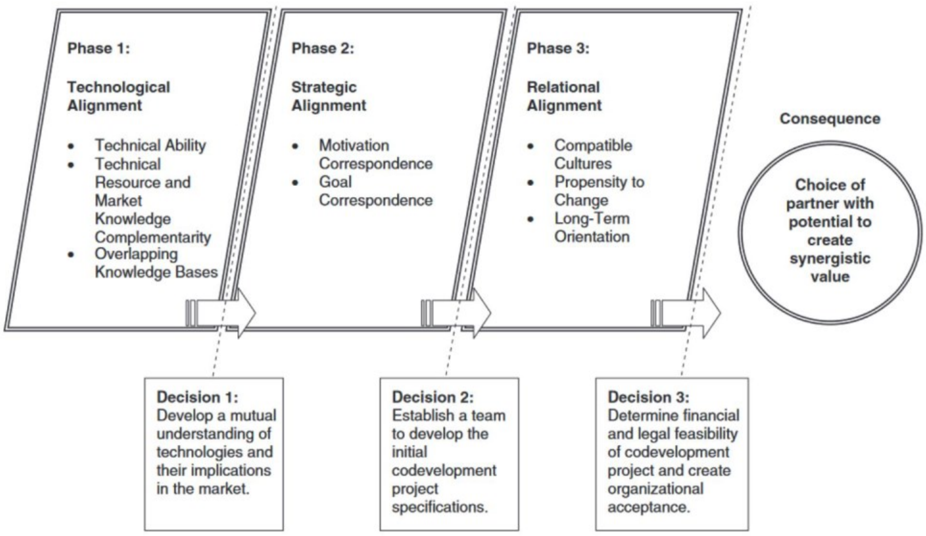

Figure 5: Model 2 – Co-development partner selection model (Emden, Calantone & Droge, 2006)

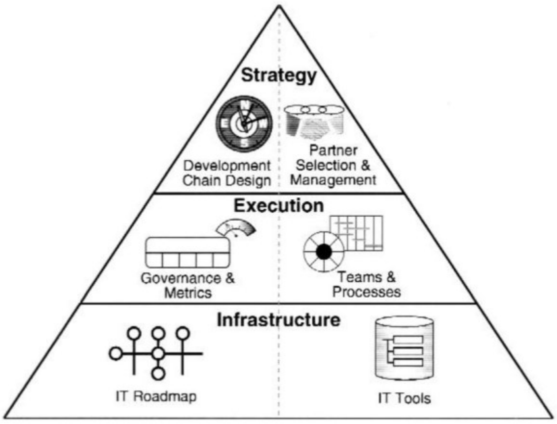

Figure 6: Model 3 - CoDev Framework [33]

Table 3: Differences and similarities in factors covered by three co-development models

Factors covered	Model 1 by Chesbrought & Schwartz (2007)	Model 2 by Emden, Calanton and Droge (2006)	Model 3 by Deck and Strom (2002)
Asses missing inhouse capablities	x		
Complementary technological capbalities	x	x	
Defining common goals / business alignment	x	x	x
Equal efforts of parties / financial responsibility	x	x	
Long term orientation / flexibility to adapt	x	x	
Compatible cultures		x	
Stages in formation proces		x	
Team formation		x	x
IT infrastructure			x

Figure 7: All 3 Models compared [33]

The three different models do not offer a unified vision of co-development. However, some of the points are mentioned in at least two of the three models. They are therefore considered as predominant for the good implementation of co-development:

- Complementarity in technological capabilities
- Sharing of common goals and business alignment
- Balance of the effort (financial point of view)
- Long-term orientation and flexibility to adapt
- Establishment of a team

Co-development best practices

The following chapter analyses in more detail the best practices and success factors that we have been able to gather from various exchanges and from the literature.

Two types of success factors are considered before looking at best practices: hard and soft success factors.

The hard success factors are:

- Realization of goals
    - Set your goals: The product vision and goals should be the mutual aim. It´s crucial and valuable to set goals related to delivering value to the user as well as budget and time goals. [34]
- Realization of service levels

    e.g.: achieve a 2-hour resolution mean time by sharing our resources and efforts)

- Expansion of the scope of the contract (e.g.: we start by collaborating on a PoC and extend if the collaboration is successful)
- Renewal of the contract (e.g.: the collaboration is bearing fruit, and nobody seems interested in finding other alternatives)

The soft success factors are:

- Customer satisfaction
- Active communication
    - Communication channels: Daily contact must be as simple and accessible as possible for both parties, use the generally available tools. [34]
    - Communicate! There is nothing more important than establishing the right working relation between both partners. Keep in mind the more information is given of the products vision, the better the output [34]
    - Trust your partner, do not hesitate to speak your mind or give some suggestions.
- Involvement
- Cultural fit
    - Team roles and project management methodology: To make sure that each project member knows his/her role and responsibilities. An effective approach to project management is needed. The popular scrum may be helpful here, to ensure a clear team structure and higher efficiency. [34]
- Trust
    - Choose the right company and just trust them. Excessive control over projects may have adverse effects. Clients would often interfere with the partners work without having proper expertise on a certain technology. Such situations where the customer thinks that particular product functionalities can be created very quickly, while developers actually need a lot of time to properly integrate them. The customer must understand and accept these things. [34]

The success of a collaboration lies in the trust established between the technical teams, strong communication as well as a clear R&D roadmap defining each party’s responsibility. [31]

Recommendations for R&D leaders

The following insights from the research are considered valuable recommendation for R&D leaders:

- Standardize partnership selection criteria to better compare the strengths and weaknesses of potential partners.
- Prioritize aligning R&D's organizational goals with co-development partners' goals to ensure a fruitful and well-scoped working relationship.
- Quantify and formalize partnership evaluation criteria to understand how engagements led to successes or failures within the co-development process.
- Examine new paradigms for intellectual property (IP) valuation and management to expand co-development opportunities and capture additional value from these relationships. [35]

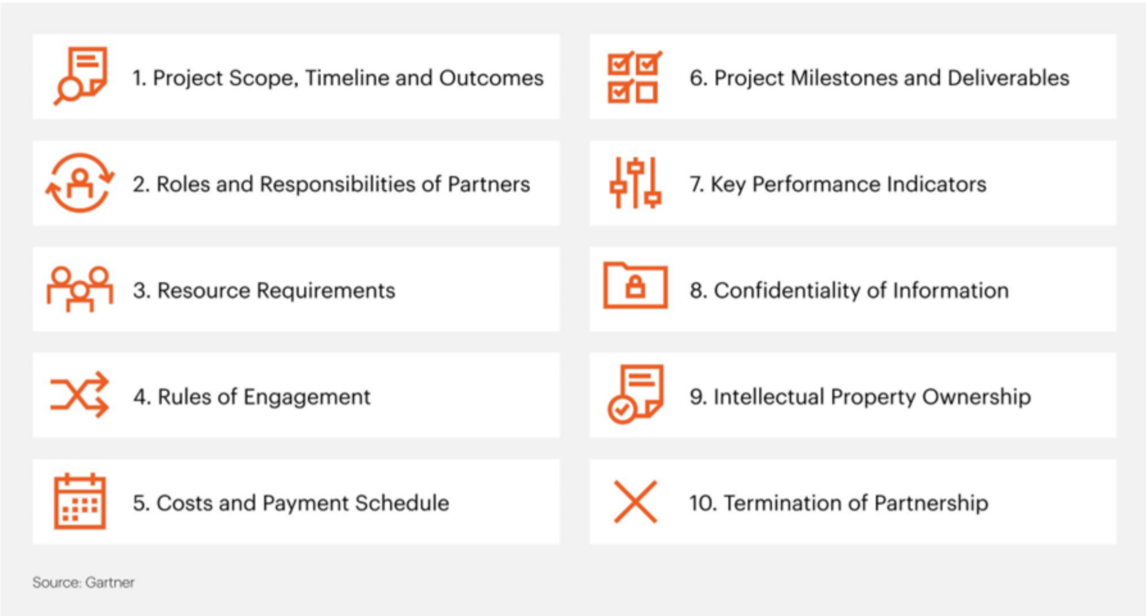

Figure 8: Check List for External Partnerships - Source: Gartner

As a result, the research identified 8 best practices to best approach a co-development project (Chan 27.05.2017)

#1 – Define and prioritize specific partnership outcomes upfront.

In today's dynamic market, speed is of the essence. It is critical to define specific outcomes, deliverables and milestones before the partnership is formally executed.

If the companies are unable to agree or commit to specific things upfront, then there is no partnership.

#2 – Plan ahead for changes in the partnership.

As the market and its ecosystems evolve, so will partnerships. The nature of the changes in the market will drive how the partnerships are affected. Some partnerships will go away, while others need to be rescoped.

Plan for changes in the partnership by reviewing the partnership programs on a quarterly basis. Update and adjust the partnership types, relationships and scope. Revisit the specified outcomes and priorities to determine what changes are necessary. When creating partnership agreements, specify how changes to the original partnership, including terminations, are handled. Avoiding overcommitting and locking into a partnership arrangement that is hard to extract from.

#3 – Strive for quality of partnerships, not quantity.

It is critical for companies to partner selectively and not over-partner. It is more important to commit to a small number of partnerships with strong joint value and get things done, than many partnerships that drain valuable resources and accomplish nothing.

#4 – Drive accountability across all levels.

Partnerships are often crafted at the executive levels but left to the field or working levels to implement. In the fast-moving market where changes happen in days and weeks, instead of months and years, it is critical that all levels are equally invested and accountable for the success of the partnership. Partnership relationship managers, on both sides, are accountable for the overall management and execution of the partnership. Executives are responsible for continuously reviewing, prioritizing and committing to the partnerships. At the execution levels, managers are accountable for staffing, prioritizing and implementing the partnership outcomes and goals. This accountability is connected across all levels, from top down, to ensure partnership success.

#5 – Build and maintain relationships at all levels.

Joint value is what brings partners together. But good relationships are the glue that holds the partnerships together on a day-to-day level. Mistrust, personal conflicts and self-interest will render the most strategic of partnerships ineffective. Good relationships can get things done and done faster. Good relationships can make regular partnerships act like strategic partnerships. One key “must do” is to staff people with strong interpersonal, influence, and relationship building skills in critical partner management and liaison roles on both sides.

#6 – Align your solution releases with each other.

Partners that integrate with each other must align their roadmaps so that compatible products and upgrades are released concurrently. When the solutions are released out of sequence, customers will experience system compatibility issues. This ultimately results in system integrators and customers who refuse to upgrade their solutions to the latest version because they are afraid it will “break” something else.

#7 – Continuously invest in the partnership.

It is easy to become complacent once the partnership is signed or when the partnership is producing. In order for the partnership to remain effective and productive on a consistent basis, management on both sides must continuously invest in the partnership. This can be in the form of adding more resources to the partnership, developing more solutions together, pursuing new markets jointly, or rotating in top talent into the partnerships.

#8 – Develop "Ecosystems Thinking".

As new use cases emerge and product ecosystems continue to evolve, solutions vendors must look beyond their corporate boundaries to be successful. Whether they like it or not, their success depends on the success of their partners and the ecosystem. They must design for their partners and the ecosystem as much as their own needs. Successful vendors are ecosystems architects, not just solutions providers. This is by training executive and managers to become "systems thinkers", developing product managers and engineers into systems engineers, and incorporating systems engineering methodologies into their product development programs.

Dealing with IP

Intellectual property plays a crucial role in business performance, especially considering a collaboration between two or more businesses. It has become the central resource in almost all industries and therefore it demands adequate methods for its valuation.

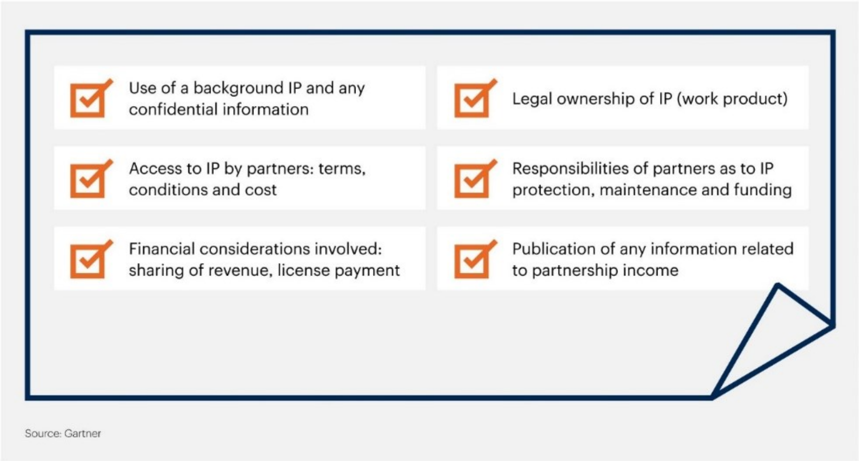

Figure 9: IP negotiations checklist [35]

Considering new approaches to intellectual property valuation and management, the following key topics can be mentioned:

- Co-development partner access to intellectual property, background intellectual property and confidential information should be subject to conditions, terms or licenses.
- Financial considerations should be involved, such as the sharing of revenue or license payment
- IP management should focus on creating a "win-win" scenario for all parties from the beginning
- Creating a non-legally binding document outlining the objectives of the partnership, in collaboration with all involved partners
    - Use this document as a guideline during the partnership, subsequent formal contracts between the company and its partner can use this document as a basis.

- Responsibilities of every partner protecting and maintaining IP during the whole development process
- Handling of proprietary information and addressing disputes that may arise
- Publication of any information related to partnership income
- IP ownership agreements

Method	When to Use	Advantages	Disadvantages
Rule of Thumb or 25% Rule Industry Standard	• Preliminary evaluation or baseline estimation of IP value • Valuation information for comparable IP and/or IP transactions available • Technology well-established	• Provides a baseline value of IP • Requires minimal time investment	• Comparable historical IP data not always readily available • Fails to adjust for risk • Lack of standardization due to subjective qualitative indicators
Cost Approach	• Projecting future earnings from IP extremely difficult • Selling the IP to a third party not immediately possible	• Enables valuation of unique IP when little secondary information exists	• Difficult to accurately quantify IP depreciation • Difficult to estimate replacement costs for unique, newly developed assets
Income Approach	• Licensing or selling patents, trademarks and copyrights	• Provides an interim measure of IP profitability • Enables estimation of IP value in the absence of market information	• Difficult to estimate IP value through a discounted cash-flow analysis • Income attributable to IP can fluctuate with business cycles • Difficult to forecast earnings for unique, newly developed IP assets
Market Approach	• Selling bundled IP • Measuring the value of an entire IP portfolio • Engaging in mergers and acquisitions	• Enables accurate valuation when companies have access to information on historical transactions involving similar IP	• Accuracy negatively impacted by company's market value fluctuations

Source: Gartner

Figure 10: Comparative analysis of IP valuation methods [35]

There are several approaches to IP valuation with their own advantages and disadvantages (see Figure 6).

The following source offers some more information about the industry standard ("25% rule"): https://www.royaltyrange.com/home/blog/what-is-the-25-rule-in-intellectual-property-valuation

The three other approaches can be applied to IP only to a limited extent. In the following document, approaches and specific methods for intellectual property valuation are examined and its applicability is critically discussed with a special interest on technical IP. Conditions, weaknesses and frontiers are shown, and practical advice is given where possible: https://doi.org/10.5539/ijbm.v7n9p40

Risks and challenges management

Handling risks during the co-development process includes identifying and assessing risks, as well as managing and mitigating those risks.

Risks arise when two entities failed to clearly define:

- the nature of their relationship
- their precise deadlines
- the IP ownership or pricing strategy
- the mutual responsibilities
  - the goals of the partnership, to identify a mutually beneficial or a win-win approach
    - making co-development partnership management and evaluation essential
  - their own limits
    - never hesitate to communicate them and seek external advise if needed

You will also find below the different challenges as mentioned by Gartner for co-development:

Challenge	Description
Missing opportunities to work with new partners due to time and resource constraints	Most organizations work with existing partners instead of evaluating new partners for co-development, since finding and assessing a partner with the right strategic and cultural fit is both difficult and overwhelming. Organizations may miss new opportunities or capabilities available because they do not have the capacity to discover and evaluate new partners.
Efficiently and equitably establishing statement of work terms	Organizations struggle with defining the objective or goal of the partnership, identifying a mutually beneficial agreement for all parties involved, and establishing milestones and KPIs to track partnership success (such as lessons learned, insights and the commercial outcomes from the partnership). Part of this challenge is coming to an agreement on "who does what" in the partnership.
Reducing the amount of time spent negotiating contract terms, especially around IP	R&D leaders often complain that their organization's legal teams can take weeks or months to write and negotiate contracts, specifically on IP ownership and confidential data sharing. This often delays the partnership and results in suboptimal development outcomes.
Effectively managing partnership performance against goals	Given the complexity of co-development projects, organizations often notice duplicate work and struggle to efficiently manage partners' resources. When working with different types of partners, it is challenging to ensure that the schedule and cadence of expected deliverables is realistic for the partner type (e.g., startup versus university timelines).
Internal challenges within the R&D organization, including navigating approval processes, managing internal stakeholders, and securing sufficient resources	Bureaucratic slowdowns often lead to project delays, especially when organizations partner with more agile startups. Additionally, stakeholder management and selection is also a challenge. Some organizations struggle to decide which internal stakeholder to include in decisions and at what stage. Finally, R&D co-development teams often report insufficient resources to effectively manage and work on these partnerships (i.e., funding, FTE time, etc.).

Commercializing partnership outcomes	Organizations struggle with sharing the information learned from these partnerships internally and identifying ways to use/commercialize partnership outcomes.
Shifting to a decentralized R&D external partnership strategy	Some leaders expressed pain points when decentralizing their external partnership strategy, including: • Duplicating work or creating inefficiencies, especially across geographies. • Inertia to change from senior leadership. • Difference in the dynamics between high technology/software and "traditional" (i.e., manufacturing) companies' approach to R&D. • Resource constraints (i.e., insufficient FTE time, budget, capacity, etc.).

Note: Taken from structured interviews with senior R&D leaders in 2021.
FTE = full-time equivalent; IP = intellectual property; KPI = key performance indicator; R&D = research and development
Source: Gartner

Figure 11: Common challenges in co-dev partnership process [35]

Conclusion

This document should have given you a clear overview of how best to approach collaboration in smart product development and a better understanding of the concept of co-development and what it entails.

Once you have a clearer idea of what needs to be developed, you will soon be asking yourself questions about "make or buy". In the case of a "make" oriented strategy, it will then be necessary to define whether you want to move towards a subcontracting/agency model, co-development or whether you want to develop on your own.

If you are interested in the concept of co-development as we describe it in this document, there are several critical points that need to be addressed for this venture to be successful. For example:

- The selection of a good partner with whom you will eventually test your compatibility on a limited project (PoC) before considering a larger scope
- The desire to achieve common goals with identified milestones
- Putting existing IP on the table and sharing the IP that will be generated
- The method of remuneration for each of the partners following the success of the co-development/IP sharing

We hope that the best practices outlined here, as well as the success factors, risks and challenges, will help you to see if you are interested in this venture.

It may be best to experiment with a small project to see what it means for your organization and then prepare for future larger co-development partnerships.

References

[1] Hicking, J., Zeller, V., Schuh, G., 2018. Goal-Oriented Approach to Enable New Business Models for SME Using Smart Products, in: Chiabert, P., Bouras, A., Noël, F., Ríos, J. (Eds.), Product Lifecycle Management to Support Industry 4.0. 15th IFIP WG 5.1 International Conference, PLM 2018, Turin, Italy, July 2-4, 2018, Proceedings. Springer International Publishing, Cham, pp. 147–158.

[2] Abramovici, M., Herzog, O., 2016. Engineering im Umfeld von Industrie 4.0: Einschätzungen und Handlungsbedarf. Acatech Studie, München, 48 pp.

[3] Porter, M.E., Heppelmann, J.E., 2014. How Smart, Connected Products Are Transforming Competition. Harvard Business Review (9), pp. 1–23.

[4] Lünnemann, P., Wang, W.M., Lindow, K., 2019. Smart Industrial Products: Smarte Produkte und ihr Einfluss auf Geschäftsmodelle, Zusammenarbeit, Portfolios und Infrastrukturen, 66 pp.

[5] IIC, 2019. The Industrial Internet of Things: Volume G1: Reference Architecture. Industrial Internet Consortium, 58 pp. https://www.iiconsortium.org/pdf/IIRA-v1.9.pdf. (Last checked: 07/27/2020)

[6] VDI/VDE-Gesellschaft. Status Report: Reference Architecture Model Industrie 4.0 (RAMI4.0), July 2015, 28 pp. https://www.zvei.org/fileadmin/user_upload/Presse_und_Medien/Publikationen/2016/januar/GMA_Status_Report__Reference_Archtitecture_Model_Industrie_4.0__RAMI_4.0_/GMA-Status-Report-RAMI-40-July-2015.pdf (last checked: 08/22/2020)

[7] ISO, 2018. Internet of Things (IoT) – Reference architecture. ISO, Genèva ISO/IEC 30141, 88 pp.

[8] Open Industry 4.0 Alliance. Über uns, 2020. Über uns. https://openindustry4.com/de/About-Us.html. (Last checked 08/16/2020).

[9] Reistad, B., 2018. Microsoft Azure IoT Reference Architecture, 71 pp.

[10] Fitzsimons, P., Carlson, B., Potter, B., Lester, R., Ewart, J., Besh, N., Steele, J., King, R., Rifkin, E., Ramsay, M., Paddock, S., Hughes, C., 2019. AWS Well-Architected Framework, 85 pp.

[11] Adaptive Manufacturing Open Solutions (ADAMOS), 2020. Maschinenbau gestaltet Digitalisierung, 8 pp. https://exhibitorsearch.messefrankfurt.com/images/original/userdata/bata/204981/5dc125b8b012b.pdf (Last checked: 08/22/2020)

[12] FIWARE, 2020. Smart Industry - FIWARE Open Source Platform for Smart Industry. https://www.fiware.org/community/smart-industry/. (Last checked: 08/22/2020)

[13] vom Brocke, J., 2015. Referenzmodellierung: Gestaltung und Verteilung von Konstruktionsprozessen, 2., unveränderte Auflage ed. Logos, Berlin, 434 pp.

[14] Krcmar, H., Reidt, A., Duchon, M., 2017. Erstellung einer Referenzarchitektur anhand von individuellen Unternehmensanforderungen, in: Bullinger-Hoffmann, A.C. (Ed.), S-CPS: Ressourcen-Cockpit für Sozio-Cyber-Physische Systeme. Abschlussveröffentlichung. aw&I - Wissenschaft und Praxis, Chemnitz, pp. 23–42.

[15] Porter, M.E., Heppelmann, J.E., 2015. How Smart, Connected Products Are Transforming Companies. Harvard Business Review (10), pp. 96–112.

[16] Eisenhardt, K., 1989. Building Theories from Case Study Research. The Academy of Management Review 14 (4), pp. 532–550.

[17] Eisenhardt, K., Graebner, M., 2007. Theory Building From Cases: Opportunities and Challenges. Academy of Management Journal (50), pp. 25–32.

[18] Crawley, E., Cameron, B., Selva, D., 2016. System architecture: Strategy and product development for complex systems, Global Edition ed. Pearson, Edinburgh, 479 pp.

[19] Schütte, R., 1998. Grundsätze ordnungsmäßiger Referenzmodellierung: Konstruktion konfigurations- und anpassungsorientierter Modelle. Gabler Verlag, Wiesbaden, 419 pp.

[20] Schuh, G., Stroh, M.-F., Hicking, J., Kremer, S., 2022. Smart Products for Smart Production – A Use Case Overview, in: Herberger, D., Hübner, M. (Eds.), Proceedings of the Conference on Production Systems and Logistics: CPSL 2022. Hannover : publish-Ing, pp. 504–514.

[21] Wuest, T., Schmidt, T., Wei, W., Romero, D., 2018. Towards (pro-)active intelligent products. IJPLM 11 (2), pp. 154 - 189.

[22] Feldhusen, J., Grote, K.-H. (Eds.), 2013. Pahl/Beitz Konstruktionslehre: Methoden und Anwendung erfolgreicher Produktentwicklung. Springer, Berlin [u.a].

[23] Schuh, G., Kreutzer, R., 2017. Methodology to Assess the Utility Potentials of Cyber-Physical Systems' Field Data: A Literature Review and Rough Solution Concept. International Journal of Emerging Engineering Research and Technology 5 (12), pp. 1–10.

[24] Mertens, P., Barbian, D., Baier, S. (Eds.), 2017. Digitalisierung und Industrie 4.0 – eine Relativierung. Springer Fachmedien Wiesbaden, Wiesbaden.

[25] Nunes, M.L., Pereira, A.C., Alves, A.C., 2017. Smart products development approaches for Industry 4.0. Procedia Manufacturing 13, pp. 1215–1222.

[26] Dahlqvist, F., Patel, M., Rajko, A., Shulman, J., 2019. Growing opportunities in the Internet of Things. McKinsey & Company, July 22. https://www.mckinsey.com/industries/private-capital/our-insights/growing-opportunities-in-the-internet-of-things#/ (Last checked: 01/04/2023)

[27] Westermann, T., 2017. Systematik zur Reifegradmodell-basierten Planung von Cyber-Physical Systems des Maschinen- und Anlagenbaus. Dissertation, 216 pp.

[28] Schuh, G., Stroh, M.-F., Krüger, P., 2022. Designing Smart Products for Industry 4.0: An Information Systems Architecture Prototype, in: Camarinha-Matos, L.M., Ortiz, A., Boucher, X., Osório, A.L. (Eds.), Collaborative Networks in Digitalization and Society 5.0. 23rd IFIP WG 5.5 Working Conference on Virtual Enterprises, PRO-VE 2022, Lisbon, Portugal, September 19–21, 2022, Proceedings, 1st ed. 2022 ed. Springer International Publishing; Imprint Springer, Cham, pp. 194–205.

[29] Kremer, S., Mennerich, C., Stroh, M.-F., Günther Schuh, 2023. Proposing A Solution For A Self-Managed Data-Ecosystem In Production: Use-Case-Driven IT-OT-Integration With An Event-Driven IT-Architecture, in: Conference on Production Systems and Logistics 2023-2, vol. 2. publish-Ing., pp. 300–312.

[30] klaxoon. Co-development: a forward-thinking training method that calls on collective intelligence.

[31] Early Metrics, 2017. Startups and Corporates - Part 4: Co-development for a successful business relationship > Early Metrics. https://earlymetrics.com/startups-corporates-co-development-successful-business-relationship/. (Last checked: 01/04/2023)

[32] Co-development / co-creation @ Kapernikov.

[33] Freens, A., 2017. Forming co-development alliances between large enterprises together with small/medium enterprises in the smart mobility industry. https://pure.tue.nl/ws/portalfiles/portal/76092788/Master_Thesis_Arnold_Freens.pdf (Last checked: 01/04/2023)

[34] Ferens, K. Best Practices When Cooperating with a Software House. https://thecodest.co/blog/best-practices-when-cooperating-with-a-software-house/ (Last checked: 01/04/2023)

[35] Gartner Research, 2022. R&D Leader Insights: Solving Common Challenges in Co-development Partnerships. https://www.gartner.com/en/documents/4013623 (Last checked: 01/04/2023)